# Machine Learning Revealing Insights into Soil Stratification

## An Application for Dikes and Dams

L.K. Leunge

Delft University of Technology

# Machine Learning Revealing Insights into Soil Stratification

## An Application for Dikes and Dams

by

L.K. Leunge

to obtain the degree of Master of Science
at Delft University of Technology,
to be defended publicly on Wednesday January 30, 2019 at 04:30 PM.

| | | |
|---|---|---|
| Student Number: | 4141873 | |
| Project Duration: | October 2017 - January 2019 | |
| Thesis Committee: | Prof.dr.ir. M. Kok, | Delft University of Technology |
| | Dr. B.E. Zuada Coelho | Deltares |
| | Dr. P.J. Vardon, | Delft University of Technology |
| | ir. R.E. Jorissen | Delft University of Technology and Rijkswaterstaat |
| | ir. W.J. Klerk | Delft University of Technology and Deltares |

TU Delft

# PREFACE

I would like to express my gratitude to the members of my graduation committee: Matthijs Kok, Richard Jorissen, Phil Vardon, Bruno Zuada Coelho and Wouter Jan Klerk. Their enthusiasm, commitment and guidance throughout this project have proved to be invaluable. I have enjoyed our discussions, which were educational and an inexhaustible source of inspiration.

I would like to thank Deltares, which offered me to make use of their data, software, and facilities. Also, my thanks go to their employees, from whom I received valuable advice and help anytime I was in need.

Furthermore, I would like to thank my friends. I could always count on their support and I appreciate their patience and endurance in listing to the stream of ideas and concerns regarding this thesis.

Finally, my sincere appreciation goes to my family and my girlfriend, for their great support from the start to final end of this project.

*L.K. Leunge*
*Delft, January 2019*

# ABSTRACT

In the Netherlands, robust dike and dam construction is a major concern and partially dependent on an understanding of subsoil variation. Insights into the subsoil are locally derived for example, by conducting in situ cone penetration tests or boreholes. However, the heterogeneity of the subsoil in the Netherlands, in terms of spatial variation of stratification and soil properties, drastically limits the validity range of a single test. Consequently, large structural designs like dikes and dams have to deal with inevitable spatial subsoil uncertainties. Therefore, technical requirements of dikes and dams have to incorporate a buffer that accounts for these uncertainties. This results in a conservative translation of the safety standard into cross-sectional reliability requirements. Improving the data resolution of a geotechnical analysis would decrease spatial subsoil uncertainties, which in turn would lead to more accurate technical requirements and thereby reduces the construction costs of dikes and dams. This thesis presents a proof of concept of a Machine Learning application, which, by learning locally measured information and analysing high spatial resolution surface settlement data, can increase insights into spatial variation of soil stratification below dikes and dams founded on heterogeneous subsoils, in order to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements.

The methodology used to derive insights in spatial variation of soil stratification relevant for the primary failure mechanism piping can be summarised in four steps:

1. The first step is the definition of the target, which describes the specific spatial information that is required to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements that are associated to piping. In case of piping, the target is to determine whether or not a permeable sand layer crosses the full dike or dam width.

2. The second step is the definition of the labels, which characteristics have to be chosen with respect to the set target. In case of piping, this means that the labels should at least provide a distinction between a soil stratification profile with and without a sand layer. In this thesis the stochastic subsoil scenarios that are derived for the Statutory Safety Assessment (WBI) of flood defences in the Netherlands are used as labels, which are stochastic representations of groups of similar soil stratification profiles.

3. The third step is the feature analysis. By means of the developed feature extraction method, spatial surface settlement measurements combined with information about the spatial load distribution are transformed into features (depth averaged characteristics of the soil column). By analysing the impact of the load on the subsoil scenarios, certain subsoil scenarios can directly be distinguished from the spatial distribution of features. Subsequently, the spatial overview of a dike segment can be subdivided into a classified and an unclassified part.

4. The fourth step is the application of Machine Learning. To apply Machine Learning, the target complexity has to be matched to feature representativity of the input space. Consequently, the remaining non-directly distinguishable subsoil scenarios have to be subdivided into two classes. To comply with the set target, in the composition of the classes a distinction is required between labels that contain a sand layer and ones which do not. Subsequently, the local soil stratification is classified accordingly. Based on the training data, which consists of the features and the associated labels as presented by the classified local soil stratification information, the Machine Learning algorithm (Support Vector Machine) is trained. After the training, the unclassified spatial distribution of the features can be classified. The result is a fully classified spatial overview of the dike segment, which visualises the spatial distribution of the subsoil scenarios.

The methodology has been tested on a simulated heightening case of a dike. The results of the training led to estimated prediction accuracies of approximately 75%. By analysing the spatial distribution of the subsoil scenarios, uncertainties regarding spatial variability within the length-effect factors can be reduced. Hence, it can be stated that the insights into spatial variation of soil stratification, derived by the presented Machine Learning application, can be used to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# I

# INITIATION

# 1 INTRODUCTION

Due to the presence of the Rhine-Meuse-Scheldt delta in the Netherlands, near 60% of the country's low-lying surface is prone to flooding by water from the sea, the lakes or the rivers [27]. Over its history, this continuous threat forced its inhabitants to develop a flood protection system to ensure their survival.

> *The Greek geographer Pytheas noted while passing the Rhine-Meuse-Scheldt delta around 325 BCE, that "more people died in the struggle against water than in the struggle against men"*

The result of near 2500 years of development is the world's most advanced flood control system, formed by a large interconnected network of flood defences, channels and retention areas. Figure 1.1 shows an example of a flood defence structure: the Eastern Scheldt storm surge barrier, the crowning achievement of the Dutch flood defences.

## 1.1. Situation

Despite its threats, humans have seen benefits of living in the Rhine-Meuse-Scheldt delta. The subsoil in the delta is an example of this contrast. It offers a large variety of construction materials, however, it simultaneously causes many difficulties regarding stability of structures. The upper layer of the subsoil in the Netherlands is a superposition of layers which are formed during the Holocene. In this geological era the temperature increased and the sea level rose rapidly, followed by a fall back to around its current level. On top of the glacial deposits of the last ice age, alternating marine and fluvial deposits in combination with forming of peat layers shaped a heterogeneous subsoil in the delta [51]. Problems arise in the interaction of a structure with this subsoil, causing instabilities which eventually could result in collapsing of the structure. For example, under compression by the self-weight of a structure, clay and peat layers start to consolidate. Spatial variation of layering thickness in the subsoil often lead to unequal distributed consolidation, with tilting of

Figure 1.1: The Eastern Scheldt storm surge barrier. Reprinted from *Deltapark Neeltje Jans*, n.d., Retrieved August 28, 2018, from `https://www.neeltjejans.nl/deltawerken`.

the structure as a consequence. Subjected to a water level difference and an unfavourable tilt, the maximum overturning moment of the structure could be exceeded, resulting in a tip-over of the structure. Over time, often in harsh way, one began to gain knowledge about the impact of the subsoil on structures. Although understanding of the interaction mechanisms has been increased significantly, insights into the subsoil below a structure still goes along with large uncertainties.

Insights into the subsoil are derived by e.g. conducting in situ cone penetration tests (CPT) or boreholes. In a cost-efficient way, these and complementary tests provide high quality information about the layering and the related soil properties of the subsoil at the test location. However, the heterogeneity of the subsoil, in terms of spatial variation of stratification and soil properties, limits drastically the validity range of a single test. A quick glance at the 18 500 km of primary and secondary flood defences in the Netherlands, shows that 18 325 km consist of dikes and dams [41]. Performing a geotechnical analysis of the subsoil below these structures, providing a spatial data resolution that excludes the possibility of anomalies, is simply too costly. Consequently, structural designs have to deal with inevitable spatial subsoil uncertainties. The costs of these measures to cover the spatial subsoil uncertainties are substantial. Therefore during the design phase one aims to conduct a geotechnical analysis that provides a spatial data resolution, which optimises the costs of the research itself and the costs of prevention measures [40].

## 1.2. Problem Definition

In order to increase the data resolution of a geotechnical analysis of the subsoil below dikes and dams, incorporation of a supplementary high-resolution data source is required. Recently, technological development made it possible to map surface settlement of dikes and dams in high spatial resolution (grid cells of 0.25 m$^2$), through fast and inexpensive airborne and satellite measurements [42].

The measured surface settlement in any grid cell, can be seen as the response of the soil column, compressed by a vertical load. Being able to derive information about the soil column below each grid cell from this measured response given information of the associated load, will contribute in decreasing the inevitable spatial subsoil uncertainties that are faced today. However, a method to perform this derivation is lacking, leaving the potential of high spatial resolution surface settlement data through airborne and satellite measurements unexploited.

## 1.3. Opportunity

Since the recent explosion in the amount of accessible data, growing computational capacity and affordable data storage, Machine Learning has gained fresh momentum lately. It offers opportunities to deal with highly complex problems, regarding the quantity of data involved, that are either impractical to solve each time by inventing specialised algorithms or not completely understood yet. According to Abu-Mostafa et al. [3], arguments to consider a Machine Learning application as a solution to a problem, should include the following findings:

- A data pattern exists.
- The problem is either impractical or not completely understood.
- There is available data.

If a Machine Learning could be a solution for the problem which is stated in this thesis, these requirements have to be satisfied. Since surface settlement is the response of the compressed soil column, it can be stated that a pattern should exist. Due to the dependency on too many features, back analysis of settlement curves to derive the related stratification and soil characteristics is perceived as difficult, which meets the second requirement. As settlement monitoring is a common tool in the construction of dikes and dams, availability of data is ensured. In conclusion, all requirements are met and therefore it can be stated that the application of Machine Learning might be feasible in finding a solution to the defined problem.

## 1.4. Research Framework

### 1.4.1. Objective

Regarding the problem stated in section 1.2, the following research objective has been being defined:

*Providing the proof of concept that a Machine Learning application, by learning locally measured information and analysing high spatial resolution surface settlement data, can increase insights into spatial variation of soil stratification below dikes and dams founded on heterogeneous subsoils, in order to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements.*

### 1.4.2. Research Questions

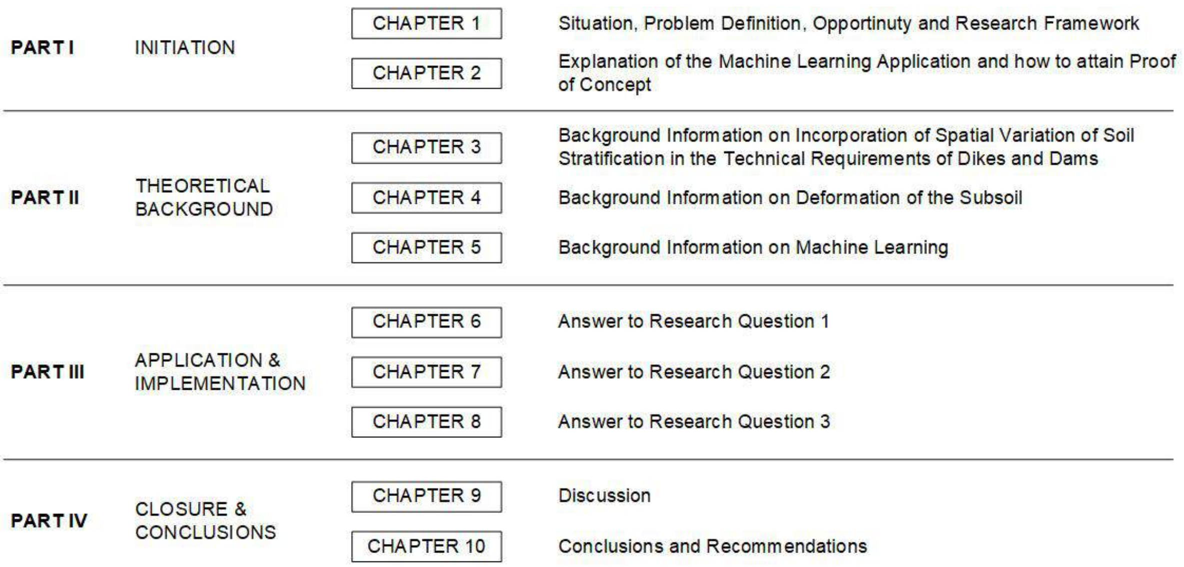
1. *How can a combination of surface settlement information and associated load information be transformed into Machine Learning features?*
2. *How is the prediction accuracy of the Machine Learning application influenced by the quantity of the training data and the target complexity of the input space?*
3. *How are insights into spatial variation of soil stratification derived and how can they be used to improve the approach of deriving technical requirements of dikes and dams?*

### 1.4.3. Research Scope

The research scope limits itself in obtaining insights into soil stratification below dikes and dams founded on soft heterogeneous subsoils, provided that the data resolution of existing local stratification is sufficient to be quantified in relevant subsoil scenarios.

### 1.4.4. Readers Guide

This thesis is divided into four parts, each subdivided into multiple chapters. Figure 1.2 shows the outline of the report.



| | | | |
|---|---|---|---|
| **PART I** | INITIATION | CHAPTER 1 | Situation, Problem Definition, Opportinuty and Research Framework |
| | | CHAPTER 2 | Explanation of the Machine Learning Application and how to attain Proof of Concept |
| **PART II** | THEORETICAL BACKGROUND | CHAPTER 3 | Background Information on Incorporation of Spatial Variation of Soil Stratification in the Technical Requirements of Dikes and Dams |
| | | CHAPTER 4 | Background Information on Deformation of the Subsoil |
| | | CHAPTER 5 | Background Information on Machine Learning |
| **PART III** | APPLICATION & IMPLEMENTATION | CHAPTER 6 | Answer to Research Question 1 |
| | | CHAPTER 7 | Answer to Research Question 2 |
| | | CHAPTER 8 | Answer to Research Question 3 |
| **PART IV** | CLOSURE & CONCLUSIONS | CHAPTER 9 | Discussion |
| | | CHAPTER 10 | Conclusions and Recommendations |

Figure 1.2: Visual representation of the thesis outline.

# 2

# IMPLEMENTING MACHINE LEARNING

An effective method to derive information from the measured response of a loaded soil column is lacking. This chapter presents a Machine Learning application that is capable of deriving soil stratification information from surface settlement measurements, making it possible to exploit high resolution surface settlement data. Firstly, in section 2.1 the implementation of the proposed Machine Learning application is conceptually shown. Secondly, in section 2.2 the required components are described and, lastly, in section 10.2, the methodology and the heightening case simulation which is used to provide the proof on concept is explained, along with the corresponding assumptions and limitations.

## 2.1. The Machine Learning Application

In this section the implementation of the proposed Machine Learning application is conceptually explained through a simulation of a fictitious dike segment that has been heightened and monitored for 1000 days. Firstly, the data known after finalising the heightening case is explained in subsection 2.1.1. Secondly, in subsection 2.1.2 the initial data complemented by the surface settlement measurement data is schematised. Thirdly, in subsection 2.1.3 the data used to train the Machine Learning algorithm is distinguished from the schematised data and lastly in subsection 2.1.4 the expected outcome of the implementation is discussed.

### 2.1.1. The Initial Data

The first step is to schematise the required information to apply the Machine Learning algorithm. The required information includes the following elements, which are visualised in Figure 2.1:

- Local soil stratification information and the corresponding spatial coordinates.
- Spatial load distribution of the old dike body and the newly applied soil layers.

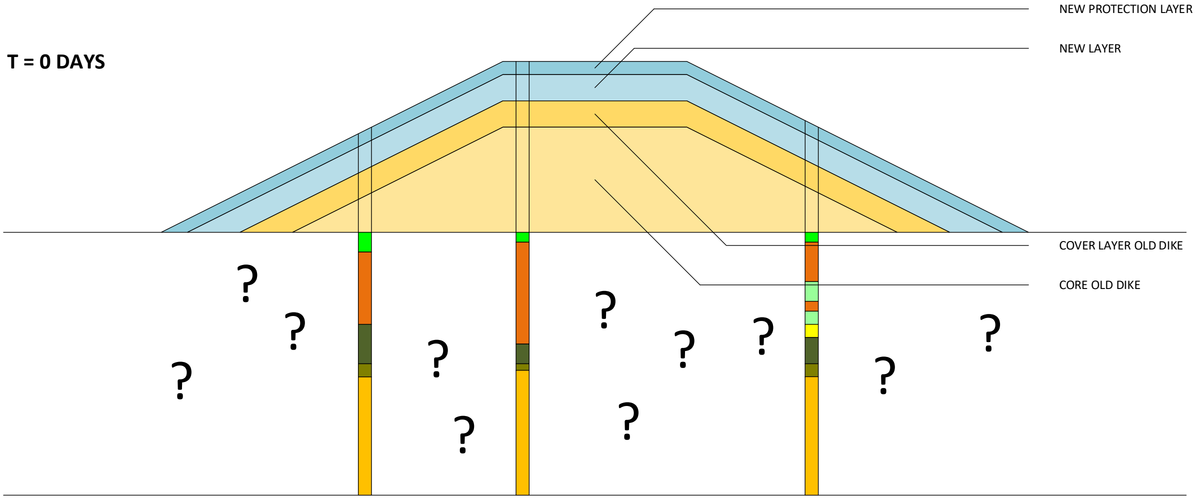


Figure 2.1: Schematised cross-sectional view of a heightened dike, including the available local subsoil information.

Figure 2.1 shows a schematised cross-sectional view of the heightened dike including the initial data. The old dike body and the new layers are coloured by yellow and blue respectively. By means of analysing previous designs of the dike, the distributed load of the old dike body is known or can be estimated. Due to usage of machinery in the execution of the heightening case, the distributed load of the new soil layers can be accurately estimated. Soil stratification information is obtained by cone penetration tests and borehole measurements. They provide insights into the soil stratification of the subsoil at a single point, but they cannot provide information about the spatial variation of soil stratification.

### 2.1.2. Schematising the Available Data

After heightening, the soft heterogeneous subsoil starts to consolidate under the weight of the newly applied soil layers.[1] Shown in Figure 2.2, by monitoring of the vertical movement of surface over time in a two-dimensional grid with a resolution of 0.25 m$^2$, through airborne and satellite measurements, the settlement of the surface can be mapped accurately. Additionally, settlement of the existing dike body, at the locations of the CPTs and boreholes, are measured by settlement rods.

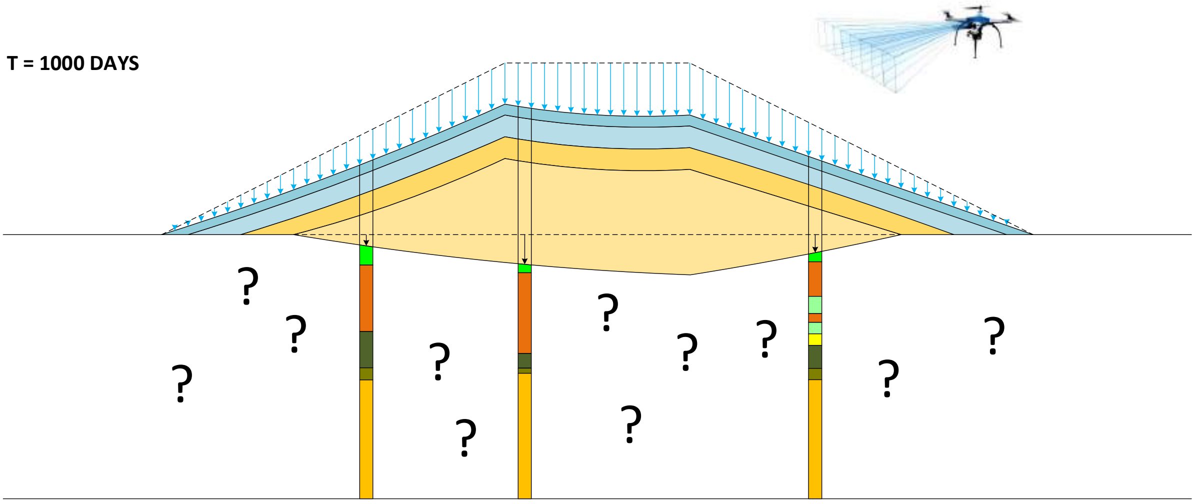


Figure 2.2: Schematised cross-sectional view of a dike heightening case 1000 days after it has been finished, including the measured response of the subsoil through surface settlement data and settlement plates.

Over time the settlement curves will capture the full consolidation process. The moment the settlement curves contain sufficient information the available data can be schematised.[2] To each settlement curve the associated load information is assigned. In combination with the execution schedule this information can be presented by a loading curve. After conducting this step, the data required to train the Machine Learning algorithm and to make predictions regarding spatial soil stratification is complete.

### 2.1.3. Distinguishing the Training Data

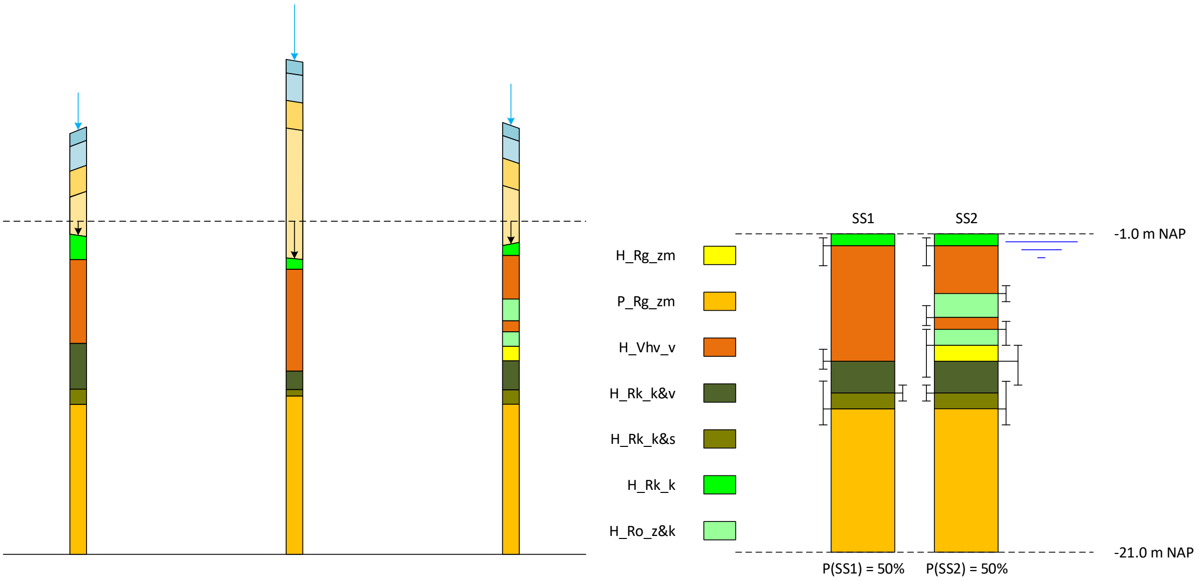
From the schematised data the training data is selected, which will be used to train the Machine Learning algorithm. The training data consists of the grid cells that contain besides a loading and settlement curve also local measurements. An example is shown in Figure 2.3a. The information contained by each of these so called "training data points" contains is:

- a loading curve;
- a surface settlement curve;
- a dike-subsoil interface settlement curve;
- a soil stratification profile.

In this proposed Machine Learning application the training data is labelled. This means that training data contains user defined prediction results. Consequently, a Supervised Learning problem is created.[3] In order

[1] More information regarding consolidation is presented in section 4.1.
[2] In section 7.3 is explained how to determine if the settlement curves contain sufficient information.
[3] In subsection 5.1.2 the concept of Supervised Learning is explained in detail.

to solve the learning problem, during its training the Machine Learning algorithm will search for the best pattern that can be detected in the training data in order to predict the correct labels. In Figure 2.3b an example of the labels that are assigned to training data is shown. The labels are in shape of the stochastic subsoil scenarios that are derived for the Statutory Safety Assessment (WBI) of flood defences in the Netherlands. These subsoil scenarios describe the estimated spatial soil stratification based on local soil stratification information. Using such labels, it becomes possible to simplify the learning problem. Instead of predicting a specific soil stratification profile, a collection of soil stratification profiles represented by a subsoil scenario is predicted.[4] Since the subsoil scenarios are derived from local soil stratification information, each of the soil stratification profiles fits a certain subsoil scenario. For example, in the example shown in Figure 2.3 it can be seen that the two left training data points of Figure 2.3a fit subsoil scenario 1 (SS1), presented in Figure 2.3b.



(a) Example of training data points.

(b) Example of labels.

Figure 2.3: Visualisation of the training data and the corresponding labels.

After assigning the labels to the training data, the grid cells of the two-dimensional grid can be subdivided into labelled and non-labelled grid cells. An example is shown in Figure 2.4. From a top view of the dike segment it can be seen whether a grid cell is classified by a certain label (SS1 or SS2) or not. Subsequently, the Machine Learning algorithm can be trained in order to predict a label for each of non-labelled grid cells.

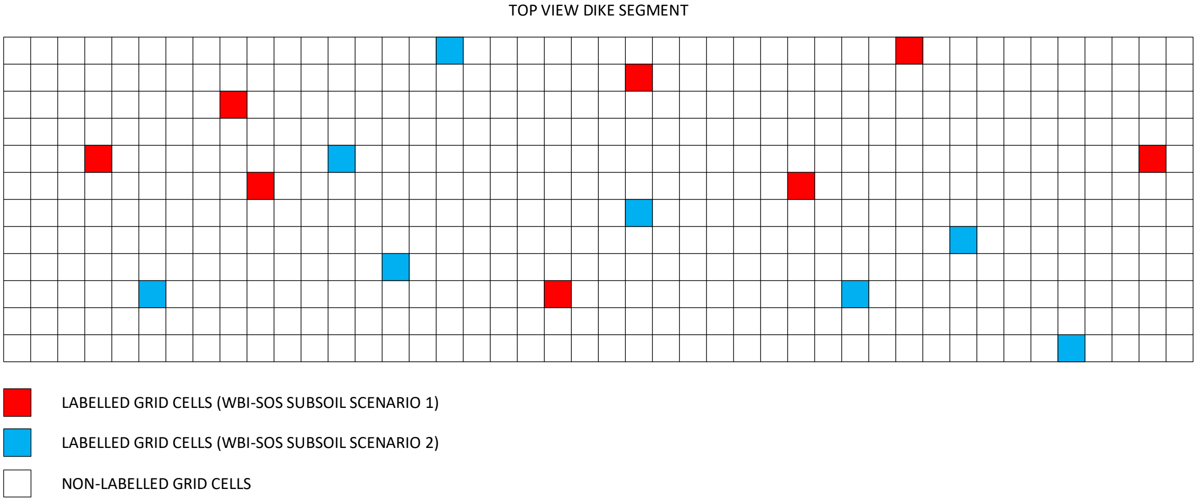

TOP VIEW DIKE SEGMENT
LABELLED GRID CELLS (WBI-SOS SUBSOIL SCENARIO 1)
LABELLED GRID CELLS (WBI-SOS SUBSOIL SCENARIO 2)
NON-LABELLED GRID CELLS

Figure 2.4: Top view of the dike segment in which the local soil stratification information is classified in terms of subsoil scenario labels.

[4] The reasons that underpin this simplification are explained in section 7.3.

### 2.1.4. The Expected Outcome

After the Machine Learning algorithm has been trained on the training data, it is capable of predicting a label for all non-labelled grid cells. In Figure 2.5 an example of such a prediction is shown. Based on the spatial information presented by the collection of loading and settlement curve in the non-labelled grid cells, the Machine Learning algorithm has assigned labels of subsoil scenario 1 and 2. In contrast to the traditional use of the WBI-SOS subsoil scenarios, which only provides a probability of occurrence of each subsoil scenario for the whole dike segment, the obtained overview presents the spatial distribution of the subsoil scenarios. Given this information the probability of occurrence of each subsoil scenario can be updated (or strengthened in case the WBI-SOS estimations are of equal magnitude) and insights into spatial variation of soil stratification can be derived.

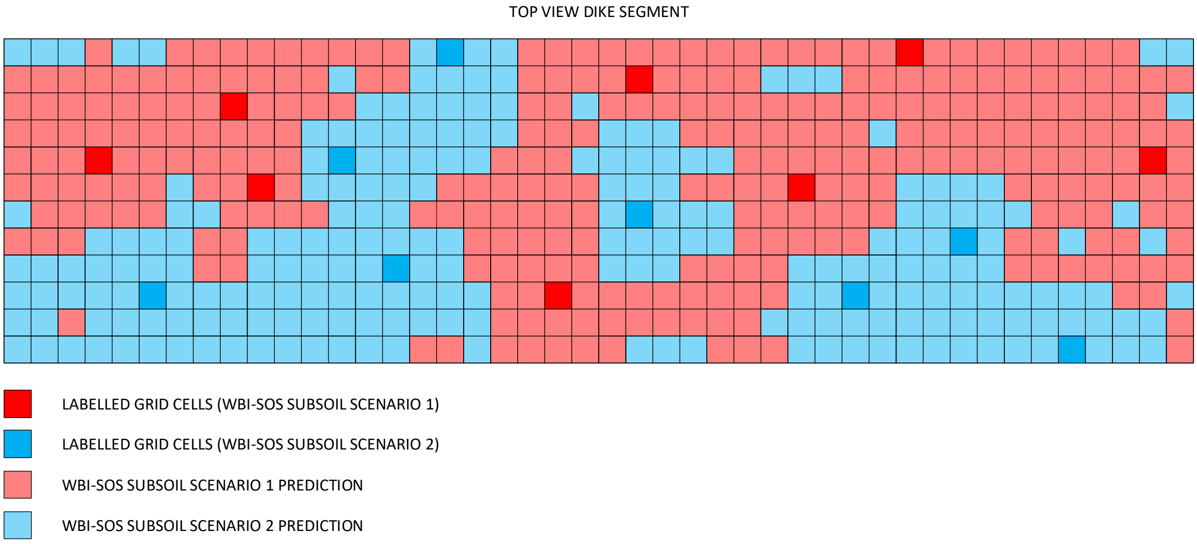

TOP VIEW DIKE SEGMENT
LABELLED GRID CELLS (WBI-SOS SUBSOIL SCENARIO 1)
LABELLED GRID CELLS (WBI-SOS SUBSOIL SCENARIO 2)
WBI-SOS SUBSOIL SCENARIO 1 PREDICTION
WBI-SOS SUBSOIL SCENARIO 2 PREDICTION

Figure 2.5: Top view of the dike segment in which all grid cells have been classified in terms of subsoil scenario labels.

## 2.2. The Machine Learning Components

In order to make a prediction given a combination of load and settlement curves, a so called feature extraction method needs to be developed and a learning model needs to be constructed. In this section the concepts of both components are briefly explained.

### 2.2.1. The Feature Extraction Method

The task of the feature extraction method is to extract features of the corresponding soil column from any combination of loading and settlement curves. Features are depth average characteristics of the soil column.[5] For example, a depth average coefficient of compressibility. In order to find these features, the feature extraction method should contain the following two abilities. Firstly, due to consolidation of the old dike body and newly applied soil layers, the measured surface settlement is larger than the measured dike-subsoil interface settlement by the settlement rods. The feature extraction method needs to be capable of filtering the consolidation of the old dike body and the newly applied soil layers from the measured surface settlement data in order to present purely the response of the compressed subsoil. Secondly, the response of the soil column is load dependent. To obtain training data that only reflects variation of soil stratification, the characteristics should be standardised with respect to the load. Hence, the second ability of the feature extraction method is to perform this load standardisation.

### 2.2.2. The Learning Model

The task of the Machine Learning algorithm that has to solve a Supervised Learning problem, is to find the pattern that correctly classifies the characteristics that are extracted by the feature extraction method. This is done by training on the labelled training data, wherein it tries to discover the structure of the training data. The most important ability of the Machine Learning algorithm is whether its predictions can be generalised. This means that the pattern needs to be found which reveals the variation of soil stratification in the whole

[5] Features are not necessarily related to physical properties.

dike segment and not only what is represented by the training data. A learning model ensures generalised predictions are obtained by using a fixed framework in which a Machine Learning algorithm is trained and the training results are validated by an error measure.

## 2.3. Proof of Concept

The goal of this thesis is to provide a proof of concept of the proposed Machine Learning application. To obtain proper understanding of the proposed method and its application domain within the Dutch flood control philosophy, the focus lays in the description of the Machine Learning processes, the associated techniques and the possible implementation of the expected outcomes. Therefore, the calculations shown in this thesis are of limited difficulty, though capable of scaling. Consequently, the used Machine Learning technique itself cannot do full justice to show its full potential. This section presents the methodology used to derive this proof of concept and the case study, along with the corresponding assumptions and limitations.

### 2.3.1. Methodology

Given the objective to provide the proof of concept of the proposed Machine Learning application, the following steps are performed:

1. Creation of the feature extraction method, through which characteristics of the compressed soil column, given a settlement curve and the corresponding load information, can be obtained.
2. Generating synthetic surface settlement measurements. Subsequently, these settlement curves and their corresponding load information are transformed by the feature extraction method to obtain the environment. The environment is the collection of all possible subsoil characteristics that can statistically be expected within the simulated heightening case.
3. Constructing the learning model for the learning process of the Machine Learning application.
4. Sampling training data sets from the environment and, subsequently, the training of Machine Learning algorithms on these training data samples by means of the learning model.
5. Mapping the quality and the capacity of the training through measuring the accuracy in reconstructing the environment.

### 2.3.2. Heightening Case Simulation

In this thesis a fictitious heightening case of WBI-SOS segment 15022 is assumed. The segment is located at the northern side of the river Lek, between the towns of Bergambacht and Schoonhoven as shown in Figure 2.5. The choice for this segment in this thesis has no scientific motive, but is of practical reason. During the initiation phase of this thesis the water authority Schieland en de Krimpenerwaard offered data from a recent monitoring survey of segment 15022. In order to present a proof of concept of the proposed method, usage of the data was not immediately required. However, with the prospect to validate the method, monitoring data is necessary. Therefore, it was decided to use the subsoil scenarios of segment 15022 are used in this research.

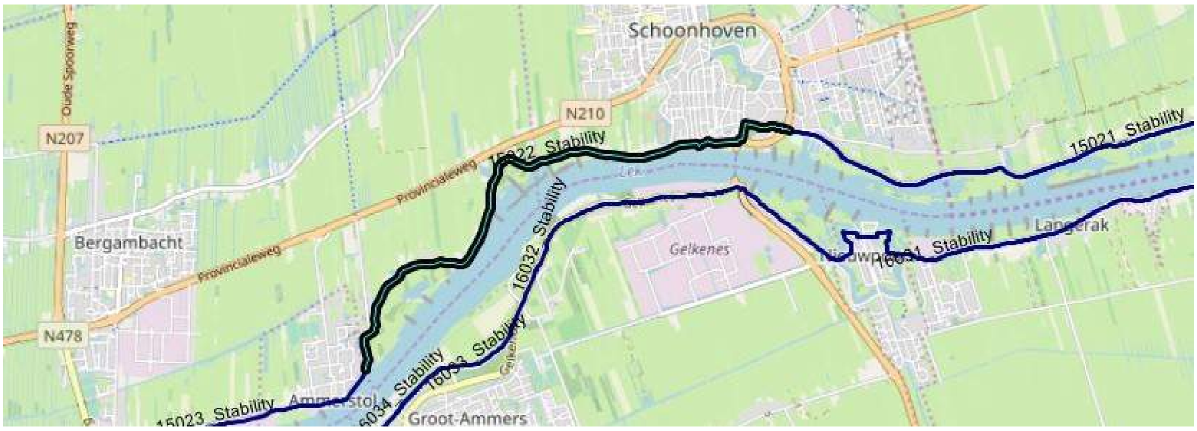


Figure 2.6: Location of WBI-SOS segment 15022. Adjusted from D-Soil Model.

It is assumed that over the full length of the segment, approximately 4.5 km, the dike prior to the heightening phase consists of a fully consolidated core of mixed local soils (18 kN/m$^3$) with a thickness of 4.0 m, covered by a fully consolidated clay layer (19 kN/m$^3$) with a thickness of 1.0 m. During the heightening phase, over the full length of the segment, the old dike is covered with a sand layer (20 kN/m$^3$) of 1.0 m and finally protected by a clay layer (19 kN/m$^3$) with a thickness of 0.5 m. Both inner and outer slopes were assumed to be 1 : 2 before the heightening case and will remain 1 : 2 after the heightening case. Figure 2.6 presents an overview of the fictitious heightening case.

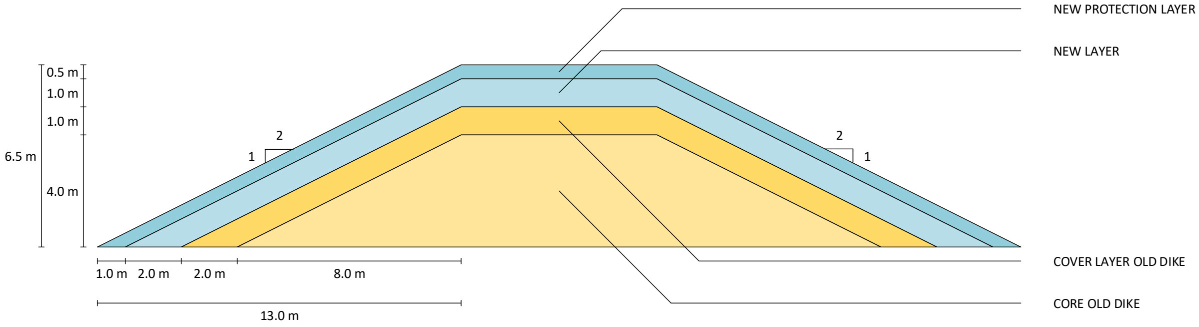


Figure 2.7: Fictitious heightening case.

The simulated subsoil below segment 15022 is represented by the corresponding stochastic subsoil scenarios that are derived for the Dutch statutory safety assessment of flood defences in the Netherlands (WBI), shown in Figure 2.7. Each subsoil scenario is expressed in terms of WBI-SOS units, of which the layer interfaces in between are defined by distributions (each characterised by a median, the 10th and the 90th percentiles), and contains an estimated probability of occurrence.

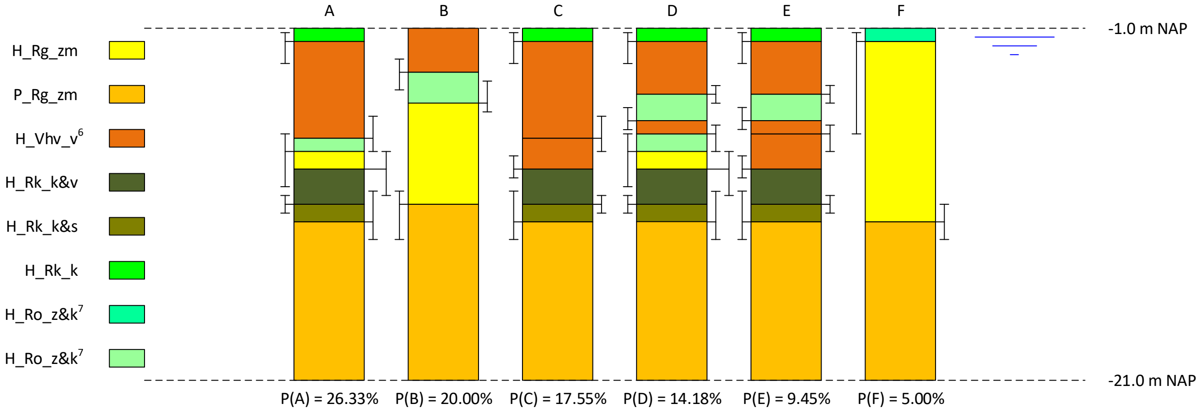


Figure 2.8: WBI-SOS subsoil scenarios associated to WBI-SOS segment 15022.[6] [7]

Unfortunately, the WBI-SOS units do not contain information of any soil parameters. Therefore, a general oedometer test data set provided by Deltares, containing data collected from all over the Netherlands, is used to assign soil layer parameters to the WBI-SOS units. The local Overconsolidation Ratios (*OCR*) associated to each WBI-SOS unit are derived from *Dijkversterking Bergambacht-Ammerstol-Schoonhoven - Uitvoeringsrapport Grondlichamen* by Lubking [30]. In Table 2.1 an overview is presented of the assigned soil layer parameter values and conditions per WBI-SOS unit.[8]

[6] According to the WBI-SOS subsoil scenarios *C* and *E*, this WBI-SOS unit appears to have two variants, as they are presented as two individual layers. However, in D-Soil Model they contain the same legend label. Since no literature defines any difference, in this thesis both variants are assigned with equal soil parameters.

[7] According to D-Soil Model, these WBI-SOS units appears to be different as they are presented by individual labels in the legend. However, no literature defines any difference, therefore in this thesis both units are assigned with equal soil parameters.

[8] More detailed background information regarding the WBI-SOS units is described in *Globale Stochastische Ondergrondschematisatie (WTI-SOS) voor de Primaire Waterkering* by Hijma and Lam [20].

Table 2.1: Assigned soil layer parameter values and conditions per WBI-SOS unit.[9]

| | | $\gamma_{\text{saturated}}$ | $a$ | $b$ | $c$ | $k$ | $e_0$ | $\sigma_p$ | OCR |
|---|---|---|---|---|---|---|---|---|---|
| **Unit** | **Drained** | [kN/m$^3$] | [-] | [-] | [-] | [m/s] | [-] | [kPa] | [-] |
| H_Rg_zm | Yes | 20.82 | $2.11 \cdot 10^{-3}$ | $1.43 \cdot 10^{-2}$ | $4.16 \cdot 10^{-4}$ | $8.53 \cdot 10^{-9}$ | 0.54 | 483.90 | 1.60 |
| P_Rg_zm | Yes | 20.82 | $2.11 \cdot 10^{-3}$ | $1.43 \cdot 10^{-2}$ | $4.16 \cdot 10^{-4}$ | $8.53 \cdot 10^{-9}$ | 0.54 | 483.90 | 1.60 |
| H_Vhv_v | No | 12.15 | $1.65 \cdot 10^{-2}$ | $2.44 \cdot 10^{-1}$ | $1.38 \cdot 10^{-2}$ | $4.14 \cdot 10^{-10}$ | 3.27 | 104.57 | 1.25 |
| H_Rk_k&v | No | 14.20 | $1.12 \cdot 10^{-2}$ | $1.51 \cdot 10^{-1}$ | $8.62 \cdot 10^{-3}$ | $3.07 \cdot 10^{-9}$ | 2.40 | 60.70 | 1.60 |
| H_Rk_k&s | No | 17.35 | $5.36 \cdot 10^{-3}$ | $6.26 \cdot 10^{-2}$ | $3.24 \cdot 10^{-3}$ | $2.03 \cdot 10^{-9}$ | 1.18 | 112.53 | 1.60 |
| H_Rk_k | No | 14.92 | $3.18 \cdot 10^{-3}$ | $6.14 \cdot 10^{-2}$ | $3.05 \cdot 10^{-3}$ | $1.45 \cdot 10^{-9}$ | 5.88 | 102.02 | 1.60 |

### 2.3.3. Assumptions

The following assumptions are made regarding the design information:

- Distributed load of the old dike body is known.
- Distributed load of the new soil layer is known.
- Both the old dike body and the new soil layer do not consolidate anymore.

For most of the dike strengthening projects old design information in combination with a proper geotechnical survey should provide sufficient information to meet the first assumption. The second assumption is acceptable, regarding the technological state of the equipment used during the execution phase. Often the first part of the third assumption is also acceptable, as intervals between dike reinforcements are sufficiently long to ensure there is no more consolidation, whereas the contribution of the latter to the total settlement can be distinguished quite accurately, provided the second assumption is met.

The following assumptions are made regarding the subsoil:

- All equivalent loads act on top of the subsoil.
- The subsoil is fully consolidated at the start of the heigthening phase.
- The subsoil is fully saturated at all times.
- There is no spatial variety in soil layer parameters values.
- The WBI-SOS subsoil scenarios are the locally calibrated subsoil scenarios to target the primary failure mechanism piping, by means of local subsoil information found within the simulated dike segment.
- The subsoil can be subdivided into a collection of isolated soil columns, which do not interact with their adjacent soil columns.

The first assumption is only applicable to surface settlement measurements next to the dike body, caused by the distribution of the weight within the subsoil. In case this assumption causes to much disturbance, it can be easily avoided by restricting the predicting range to merely the subsoil directly below the dike body. The second assumption is mostly acceptable, as intervals between dike reinforcements are sufficiently long to ensure there is no more consolidation. Nonetheless, important to include in the calibration is the secondary consolidation rate at the start of the strengthening phase. The third assumption is often the case for subsoils in tidal river zones, therefore representing a large share of the implementation possibilities in the Netherlands. The fourth assumption is not realistic but used to simplify the development of the feature extraction method. The influence of spatial soil parameter uncertainties on the prediction accuracy have to be incorporated during a real implementation. This can be done through analysing the oedometer test data in a certain dike section and calculate the standard deviations of the soil parameters corresponding to each layer. The fifth assumption is realistic since it is one of the required steps that have to be performed in order to use the WBI-SOS subsoil scenarios [29]. Since subsoil interaction through shear strength and the excess pore water pressure dissipates also in the horizontal direction, the seventh assumption not realistic. In a real

[9] The soil layer parameters in this table are explained in detail in section 4.2.

case mathematical correction have to be applied in order to take these effects into account. Calibration of the synthetic surface settlement curves to measurement data can be used to obtain the order of the mathematical corrections.

The following assumptions are made regarding the synthetic surface settlement measurements:

- The synthetic surface settlement measurements are calibrated to the labelled data points.
- The surface settlement measurements represent the subsoil settlement.
- The influence of measurement uncertainties is negligible.
- The time span of the surface settlement measurements is of such length that each settlement curve is in its secondary consolidation stage.

For a fictitious case the first assumption is acceptable, however, in a real case the synthetic surface settlement measurements have to be calibrated to the labelled data points. If the assumptions regarding the design information can be met, the second assumption is acceptable. Although the technological development in airborne and satellite measurements increases rapidly and the only direct disturbance of the signals is caused by a grass layer, it is most likely small measurement uncertainties will remain. Therefore, the third assumption is not completely realistic, but will not cause significant disturbances in the development of the feature extraction method. Nonetheless, the influence of measurements uncertainties on the prediction accuracy can be incorporated in further research. Adding a random noise to the synthetic surface settlement measurements simulates a measurement uncertainty. By varying the magnitude of the random noise, the influence of measurement uncertainty can be mapped. The fourth assumption is acceptable since a settlement curve provides a clear indication point in time, which subdivides the curve in a primary and the secondary consolidation stage. An algorithm can easily detect if the synthetic surface settlement measurements contain this indication point.

The following assumption is made regarding the feature extraction method:

- The primary consolidation stage contains sufficient information to extract features that characterise the label in form of a subsoil scenario. Therefore, the feature extraction method will not be capable to extract features from the secondary consolidation stage.

The assumption is applied to simplify the development process of the feature extraction method in order to assure the method is understood correctly. Consequently, in this thesis only features are extracted from the primary consolidation stage.

The following assumption is made regarding the learning model:

- Only one classification technique for Supervised Learning is researched.

In order to get better understanding of the technique and to make sure it is correctly applied, in this thesis only one classification technique for Supervised Learning is researched.

### 2.3.4. Limitations

The following limitations are made regarding the methodology:

- The WBI-SOS scenarios do not represent the subsoil below the dike segment.

Although this limitation is not relevant in order to provide proof of concept, it will inevitably lead to failure in applying the proposed Machine Learning application.

# II

# THEORETICAL BACKGROUND

# 3

# INCORPORATION OF SPATIAL VARIATION OF SOIL STRATIFICATION IN THE TECHNICAL REQUIREMENTS OF DIKES AND DAMS

In this chapter it will become clear how in the Netherlands spatial variation of soil stratification is quantified and subsequently how it is incorporated in the derivation of technical requirements of dikes and dams. Section 3.1 explains how information regarding soil stratification is obtained from measurements and how local information could be quantified in order to form a stochastic overview, representing the expected variation of soil stratification below the dike segment. In section 3.2 is explained how technical requirements of flood defences are determined and how spatial variation of soil stratification is incorporated in the derivation of the technical requirements of dikes and dams.

## 3.1. Spatial Soil Stratification

The information provided in this section is mainly based on *Cone Penetration Testing in Geotechnical Practice* by Lunne et al. [31], *Standard Practice for Thin-Walled Tube Sampling of Fine-Grained Soils for Geotechnical Purposes* by ASTM [4] and *Globale Stochastische Ondergrondschematiatie (WTI-SOS) voor de Primaire Waterkering* by Hijma and Lam [20].

### 3.1.1. Local Soil Stratification Information

In order to derive information regarding soil stratification, in situ tests are conducted. In this subsection cone penetration tests (CPT) and boreholes are explained.

#### Cone Penetration Test

A CPT, also referred to as the Dutch Cone Test, is the most commonly used in situ testing method, developed in 1950 at the Dutch Laboratory for Soil Mechanics. The method determines soil layer parameters that can be used to derive the soil stratification. The test is performed by pushing a cone into the subsoil, whereof the tip and sleeve resistance is measured continuously. Nowadays, more advanced cones are also capable of measuring e.g. pore water pressure, temperature, shear and compression wave velocity, etc. To provide a soil stratification profile, the results can be analysed with help of the SBT chart [38]. In Figure 3.1 an example of test results analysed by the SBT chart is shown.

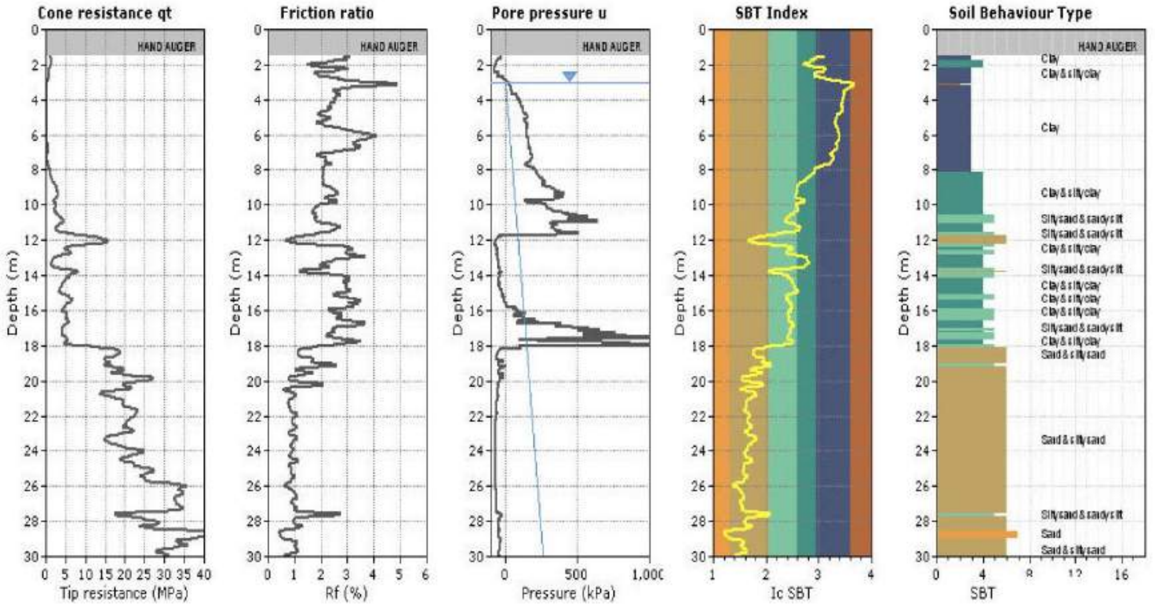


Figure 3.1: CPT results analysed with help of the SBT charts to provide a soil stratification profile. Reprinted from *Soil Behaviour Type from the CPT: an Update*, by Robertson, 2010.

**Borehole**

A narrow shaft driven or drilled in the subsoil is called a borehole. In a geotechnical survey they are used to provide soil samples for laboratory tests and provide a direct indication of the soil stratification profile at the site. The difficulty is to acquire complete undisturbed soil samples. The degree of disturbance of a sample is dictated by the dimension of the sampler and the method of recovery.[1] Although disturbances cannot be avoided, the consequences can be reduced. For example, driving methods are preferred above drilling methods and during sampling procedures in saturated soils the net hydrostatic pressure at the bottom of the bore has to be equal to zero.

### 3.1.2. Quantification of Local Soil Stratification

In this subsection an example of a soil stratification quantification method is given, which can be used to transform local information into a statistic overview. This method developed for the Statutory Safety Assessment (WBI) of flood defences in the Netherlands, and is called Wettelijk BeoordlingsInstrumentarium - Stochastische Ondergrondschematisatie, referred to as WBI-SOS. The WBI-SOS method has quantified the subsoil in subsoil scenarios for all the primary hydraulic defences, excluding the dune sections. A subsoil scenario describes a statistic soil stratification profile which is with a certain probability expected to be found within a dike segment. In Figure 3.2 the WBI-SOS subsoil scenarios for WBI-SOS segment 43047 is shown. Note, a WBI-SOS segment is not equal to a WBI dike segment.

[1] More information regarding sampling techniques can be found in *"Environmental Technology Verification Report - Soil Sampling Technology"* by Billets [10]

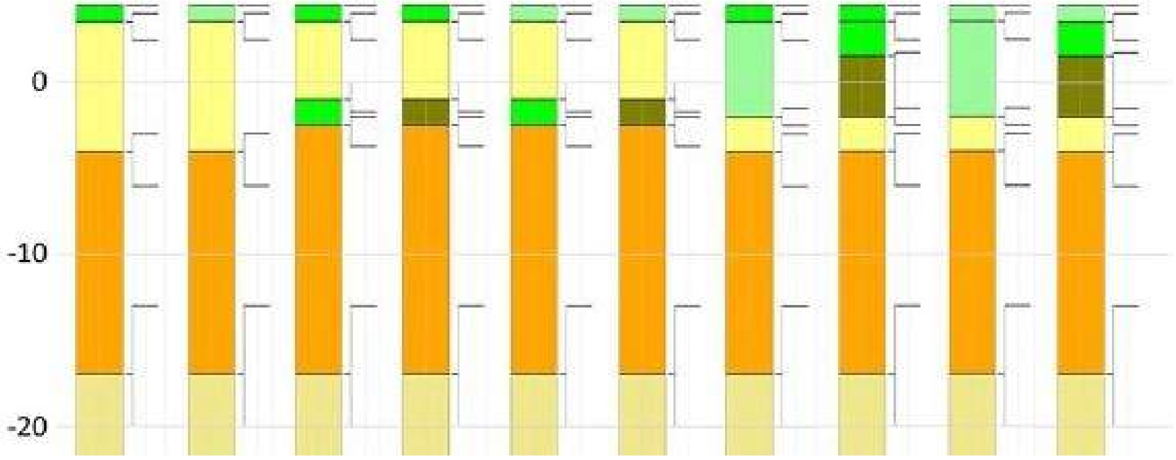


Figure 3.2: WBI-SOS subsoil scenarios for WBI-SOS segment 43047, with respectively the probabilities of occurrence of 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.04, 0.06, 0.04, 0.06. Reprinted from *Globale Stochastische Ondergrondschematiatie (WTI-SOS) voor de Primaire Waterkering*, by Hijma and Lam, 2015.

The WBI-SOS is derived in four successive steps:

1. In the first step the data has been collected. Local information was collected by CPTs and boreholes within a buffer zone of 800 m along the primary flood defences from DINOloket[2] and the database of University of Utrecht. Spatial data was obtained from three 3D geological models: GeoTOP[3], NL3D[4] and Digitaal Geologisch Model (DGM)[5]. The detailed altitude measurements from Actueel Hoogtebestand Nederland (AHN)[6] were used for the derivation of information regarding the top layer of the subsoil. Furthermore, Veiligheid Nederland in Kaart 2 (VNK2) and multiple geological maps were used to derive supplementary information.

2. In the second step the data has been organised though visualisation. With help of ArcGis[7] the location of the dike segments were visualised, information at their positions has been extracted from the 3D geological models and the local information by CPTs and boreholes was correctly placed. Subsequently, the programme iMOD[8] was used to visualise the data along the dike segments. Spatial CPT and bore hole information was perpendicularly projected at the longitudinal dike axes, up to a depth of -40 m NAP. An example of such visualisation is presented in Figure 3.3.

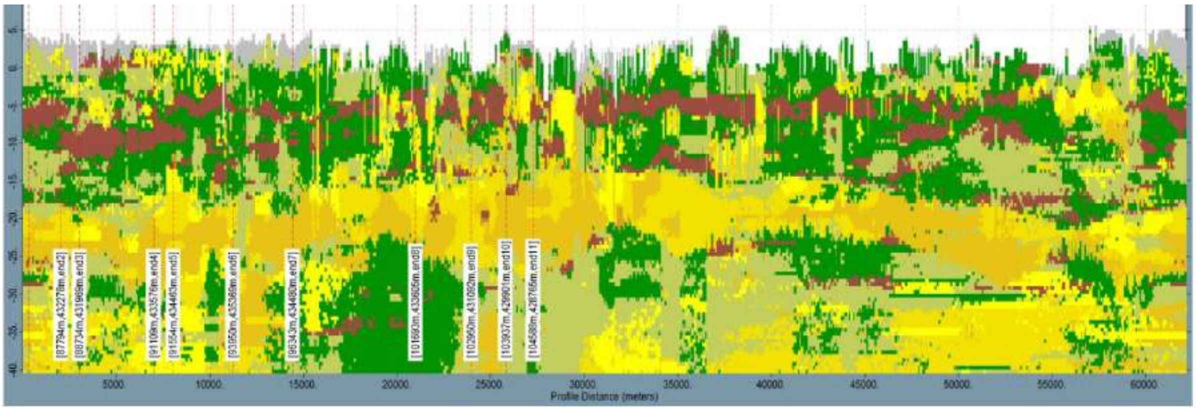


Figure 3.3: Visualisation of the collected geological data along the longitudinal axis of the dike. Reprinted from *Globale Stochastische Ondergrondschematiatie (WTI-SOS) voor de Primaire Waterkering*, by Hijma and Lam, 2015.

[2] More information regarding DINOloket can be found at: `https://www.dinoloket.nl`.

[3] More information regarding GeoTOP can be found at: `https://www.dinoloket.nl/detaillering-van-de-bovenste-lagen-met-geotop`.

[4] More information regarding NL3D can be found at: `https://www.grondwatertools.nl/nl3d`.

[5] More information regarding DGM can be found at: `https://www.dinoloket.nl/digitaal-geologisch-model-dgm`.

[6] More information regarding AHN can be found at: `http://www.ahn.nl/index.html`.

[7] More information regarding ArcGis can be found at: `http://www.arcgis.com/index.html`.

[8] More information regarding iMOD can be found at: `https://www.deltares.nl/nl/software/imod-2/`.

3. In the third step the visualised data has been analysed and WBI-SOS segments borders were drawn based on the judgment of two experts. As mentioned before, the WBI-SOS segments are derived purely from a geological perspective, and are not related to WBI dike segments. The length of the WBI-SOS segments are on average approximately 3 km, varying between a couple of hundred meters to around 15 km.

4. In the fourth step the WBI-SOS subsoil scenarios were derived, in terms of WBI-SOS units. A WBI-SOS unit is characterised by four levels of information, which need to be defined according the assessment and design guidelines of flood defences. For example, a Holocene river channel filled with fine sand is referred to as H_Rc_sf, in which river channel is a combination of two levels. The first level is the stratigraphy of the unit, e.g., Holocene or Pleistocene. The second level is the regional correlation, in terms of the deposition process, e.g., marine, fluvial, river, etc. The third level specifies the deposition environment, e.g., channel, salt marsh, lake, etc., and the last level describes the material characteristics, e.g., fine or course sand, clay, peat, etc. Based on a multi-varying visualisation of the data per WBI-SOS segment, the WBI-SOS units were assigned. This means, multiple projection of the stratification data was made for various buffer zone widths, with a maximum 800 m. From the most relevant overview the WBI-SOS subsoil scenarios have been derived. The result are stochastic soil stratification profiles with a resolution of 0.25 m, in which each layer interface is defined by median, the 10th and the 90th percentiles. Figure 3.4 shows an example of the derivation process.

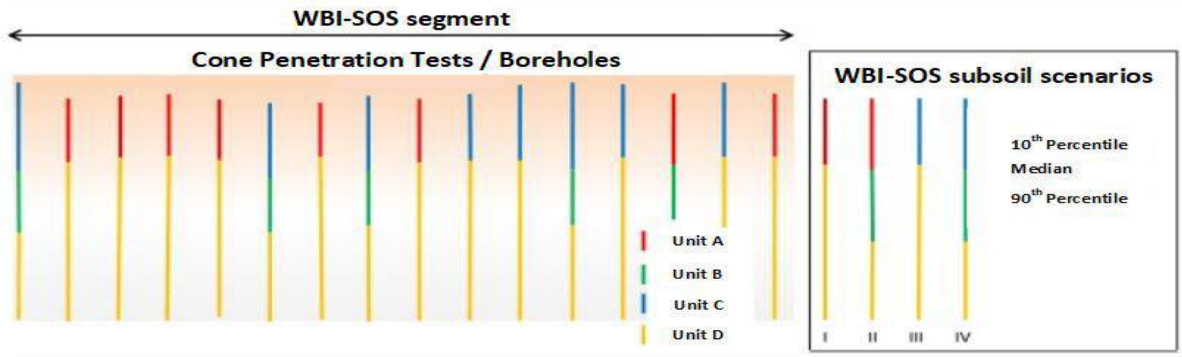


Figure 3.4: Example of the derivation of WBI-SOS subsoil scenarios. Reprinted from *Globale Stochastische Ondergrondschematiatie (WTI-SOS) voor de Primaire Waterkering*, by Hijma and Lam, 2015.

With the WBI-SOS subsoil scenarios derived, the probability of occurrence per scenario can be assigned, which is depending on the probability of occurrence per WBI-SOS unit. The probability of occurrence depends on the coFmbined assessment of two characteristics, supplemented with geological insights of experts. The first is the frequency of occurrence within the CPT and borehole data. The second is the contiguous length of a unit with respect to the WBI-SOS segment length. Lastly, per subsoil scenario the probability of occurrence per existing units are multiplied, of which the result is the probability of occurrence of the subsoil scenario within the WBI-SOS segment. Figure 3.5 shows an example of such calculation.

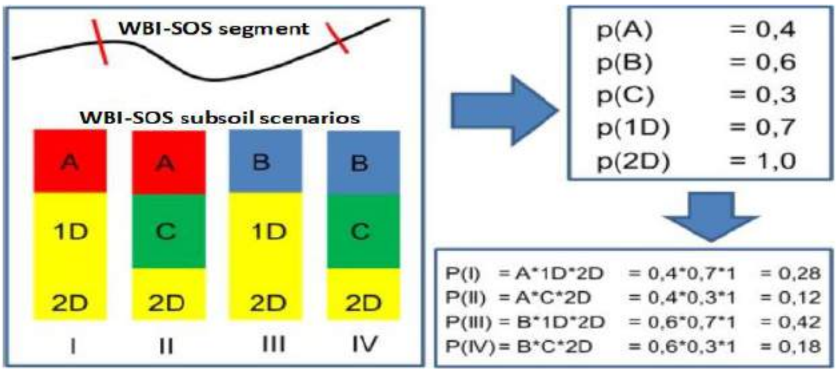

WBI-SOS segment
WBI-SOS subsoil scenarios
A
A
B
B
1D
C
1D
C
2D
2D
2D
2D
I
II
III
IV
p(A) = 0,4
p(B) = 0,6
p(C) = 0,3
p(1D) = 0,7
p(2D) = 1,0
P(I) = A*1D*2D = 0,4*0,7*1 = 0,28
P(II) = A*C*2D = 0,4*0,3*1 = 0,12
P(III) = B*1D*2D = 0,6*0,7*1 = 0,42
P(IV)= B*C*2D = 0,6*0,3*1 = 0,18

Figure 3.5: Example of determining the probability of occurrence of the WBI-SOS subsoil scenarios. Reprinted from *Globale Stochastische Ondergrondschematiatie (WTI-SOS) voor de Primaire Waterkering*, by Hijma and Lam, 2015.

## 3.2. Technical Requirements of Dikes and Dams

The information provided in this section is mainly based on *Fundamentals of Flood Protection* by Kok et al. [27], *Flood Defences* by Jonkman et al. [25], *Fenomenologische Beschrijving* by 't Hart [43] and *Safety Standards of Flood Defenses* by Vrijling et al. [50].

### 3.2.1. Flood Probability Standards

Since the introduction of the new Dutch legislation for flood defences in the Netherlands at January the 1st of 2017, a probabilistic philosophy is used to determine the technical requirements of flood defences. Whereas in the past the flood probability standards of flood defences were mainly based to be able to withstand a normative load scenario, nowadays they are derived with respect to flood risk. Flood risk incorporates besides the probability of failure given a certain load, also the consequences of a flood. The accompanying flood probability standards of flood defences are derived from an acceptable risk level in flood prone areas. The acceptable risk level is based on two principles:

- An individual is protected by a basic protection level.
- If the consequences are extremely high, the basic protection level is refined, based on societal risk and a social cost-benefit analysis.

### 3.2.2. Primary Failure Mechanisms

By definition, a flood defence has failed the moment it cannot fulfil its primary function anymore. For flood defences the primary function is to retain water. Failing in fulfilling its primary function, however, does not imply that the structure itself has collapsed. A structure can fail without collapsing, and it can collapse without failing. Also, an element of the structure can fail with or without losing of its primary function as result.

Failure in fulfilling the primary function occurs after completing a full failure mode. A failure mode is a sequence of failure of multiple sub-mechanisms. Completing a failure mode requires a certain time frame, which can be divided into stages. In Figure 3.6 an example of the failure mode terminology is visualised. The uncertainties regarding the duration of the sub-mechanisms and the moment of occurrence of stages grows if they occur closer the end of the failure mode. Therefore, it assumed that the last stage, failure of the primary function, occurs somewhere during the last sub-mechanism: breach growth.

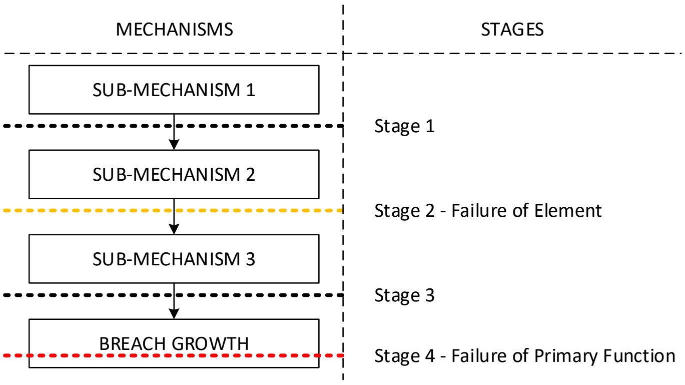


Figure 3.6: Failure mode terminology. Adjusted from *Fenomenologische Beschrijving* (p. 3), by 't Hart, 2018.

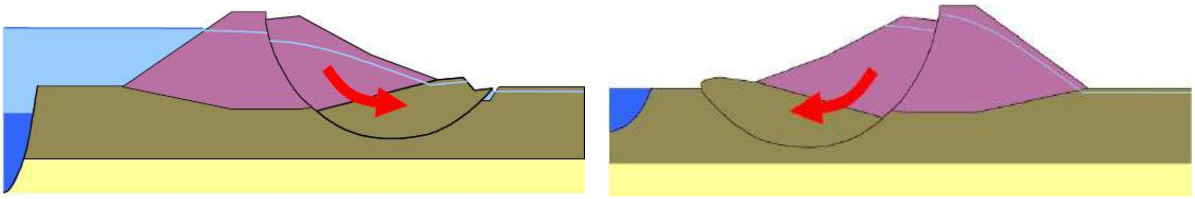
To derive the technical requirements for flood defences, failure modes have to be understood in order to determine acceptable and unacceptable behaviour of structures. For Dutch flood defences, models have been developed to perform this task. However, due to current technical limitations and the wide range of possible failure modes, models are not capable of including all existing sub-mechanisms. A consensus was yet found in defining failure definitions and primary failure mechanisms. A primary failure mechanism describes part of the failure mode up to the failure definition. Currently, the number of primary failure mechanism that have been derived cover together near 70% of the range of known failure modes. One should note, assessment based on failure definitions rather than failure modes entails that structures will be disapproved for circumstances that do not lead to failure in reality.

To conclude the terminology, a distinction is made between primary failure mechanisms that occur during a loading event and the ones that occur after. The first are called the direct primary failure mechanism and the latter indirect failure mechanisms. The importance of this distinction is to take into account the interdependencies of primary failure mechanisms that share the driving and/or counteracting forces. These interdependencies will be addressed in the formation of the failure probability budget, which will be explained in subsection 3.2.3.

To complete this subsection, a brief explanation is given, regarding the most relevant primary failure mechanisms for this thesis: macro instability (further referred to as slope stability to comply with international terminology) and piping.

### Slope Stability Inner and Outer Slope

Slope stability describes a large-scale stability problem, involving both the dike or dam body and the subsoil below. It is caused by loss of equilibrium of a soil mass due to an increase in pore water pressure, which decreases the shear resistance of the soil particles. During such events instability occurs if the driving moment exceeds the counteracting moment. If that case, the soil mass moves along a circular slip plane, resulting in large slope deformations. The occurrence of either inner or outer slope instability is often related to occurrence in time with respect to a loading event. Inner slope instability occurs more often during a loading event and is therefore called a direct primary failure mechanism. Equilibrium of the outer slope during a loading event is ensured by the water level itself. After a fast drop, this stabilising effect disappears and an unstable situation can occur. Therefore, outer slope instability mostly occurs after a loading event and is therefore called an indirect primary failure mechanism. In Figure 3.7a and Figure 3.7a respectively inner and outer slope instability are visualised.



(a) Inner slope instability.

(b) Outer slope instability.

Figure 3.7: Visualisation of inner and outer slope instability. Reprinted from *Fenomenologische Beschrijving* (p. 7 and p. 19), by 't Hart, 2018.

### Piping

In case water flows under a dike or dam through a self-eroded channel, it is called piping. It is caused by a differential in hydraulic head between the water outside and inside the structure, which starts a flow of water, also called seepage. In case of piping, seepage flows through a permeable layer in the subsoil below a dike or dam. The flow increases the hydraulic head in the permeable layer, leading to a negative differential with respect to the hydraulic head within the structure. If the water pressure by the negative differential burst the covering layer and is of such magnitude it can lead to erosion of soil particles, a channel will be formed. Piping can only occur during a loading event and is therefore called a direct primary failure mechanism. In Figure 3.8 piping is visualised.

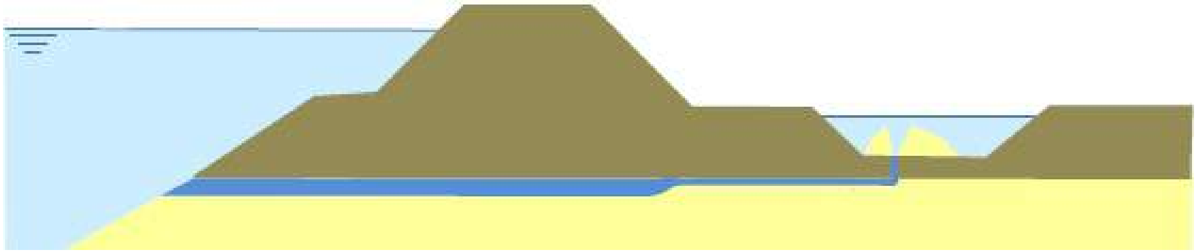


Figure 3.8: Visualisation of piping. Reprinted from *Fenomenologische Beschrijving* (p. 40), by 't Hart, 2018.

### 3.2.3. From Flood Probability Standard to Technical Requirements

With the flood probability standard and the primary failure mechanisms explained, the question rises how it is incorporated in the approach to derive the technical requirements of flood defences. The flood defences of the Netherlands are divided into dike segments. Each dike segment consists of a contiguous series of natural barriers, e.g., dunes, and/or hydraulic structures, e.g., dikes, dams, flood gates, barriers, etc. These components are distinguished such, that stochastic homogeneity of both the load and the resistance is obtained. Figure 3.9 presents an example of a dike segment, including the associated fault tree. In a series system, the system is as strong as its weakest link. The same goes for a dike segment, therefore the fault tree includes a OR gate.

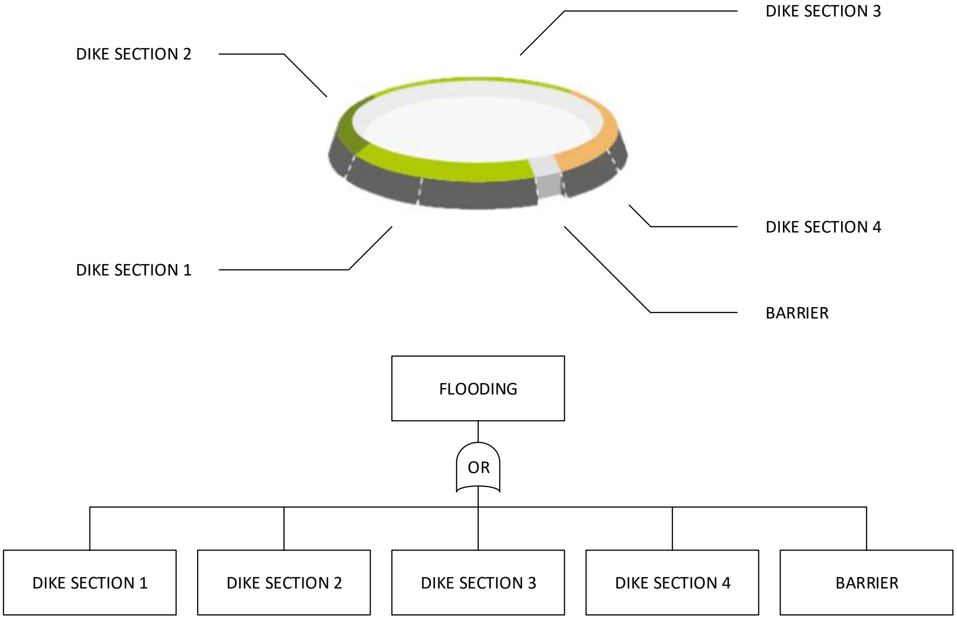


Figure 3.9: Visualisation of a dike segment and its associated fault tree. Adjusted from *Fundamentals of Flood Protection* (p. 78), by Kok et al., 2017.

It is yet not possible to determine the probability of flooding of a dike segment within a single model. Therefore, an assessment, to indicate whether the probability of failure of a primary failure mechanism exceeds the failure definition, is carried out on a representative cross-section for each component within a dike segment. These failure definitions on a cross-sectional level are called the technical requirements. In case a representative cross-section does exceed one of the requirements, the component does comply by law. In order to derive the technical requirements, the following succeeding steps have to be followed:

1. Determine a failure probability budget, given the relevant primary failure mechanisms of all dike segment components and their interdependencies.

2. Assign technical requirements to the representative cross-section per component, with respect to the permissible probability of failure per primary failure mechanism and spatial uncertainties of design parameters.

#### Determining of the Failure Probability Budget

The failure probability budget is determined to assign the contribution of each primary failure mechanism to the segment wide probability of flooding. By default, 30% of the failure probability budget is reserved for failure modes which are not yet covered by models. The remaining 70% is distributed over the relevant primary failure mechanisms according the standard failure probability budget. The standard magnitude of each failure probability budget factor ($\omega$) is derived in Vergouwe [48]. Depending on the specific situation and the interdependencies of the relevant primary failure mechanisms, for some dike segments a tailored failure probability budget has been derived in order to obtain more economic designs. Figure 3.10 illustrates an example for a fictitious failure budget, for a probability of flooding of 1/3.000 per year.

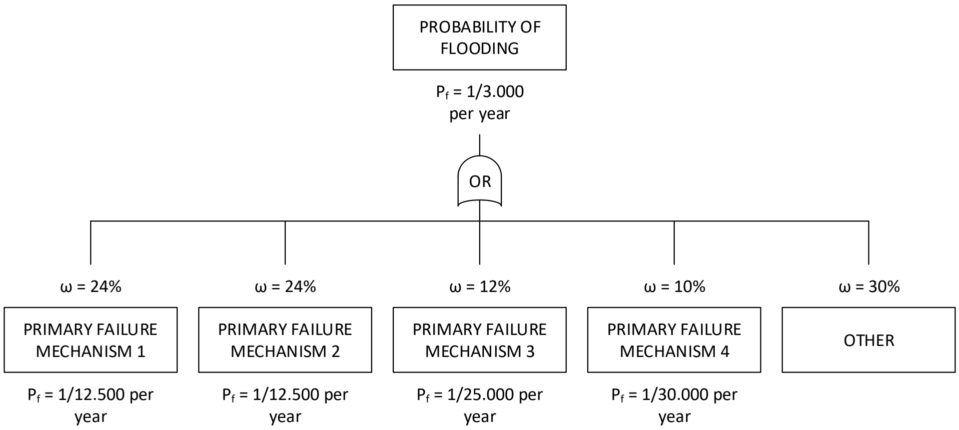


Figure 3.10: Determining a fictitious failure budget, regarding a probability of flooding ($P_f$) of 1/3.000 per year.

## Assigning the Technical Requirements

Given the failure probability budget, the contribution of each primary failure mechanism can be transformed from segment level to the technical requirements of the representative cross-section per component, as shown in Figure 3.11. To include spatial uncertainties, the transformation is conducted with help of the length-effect factors, *N*. Their origin and magnitude will be explained in section 3.3. After the transformation is completed, the technical requirements per primary failure mechanisms per component have been derived. Cross-sectional designs have to meet these requirements in order to comply with the flood probability standard of the dike segment.

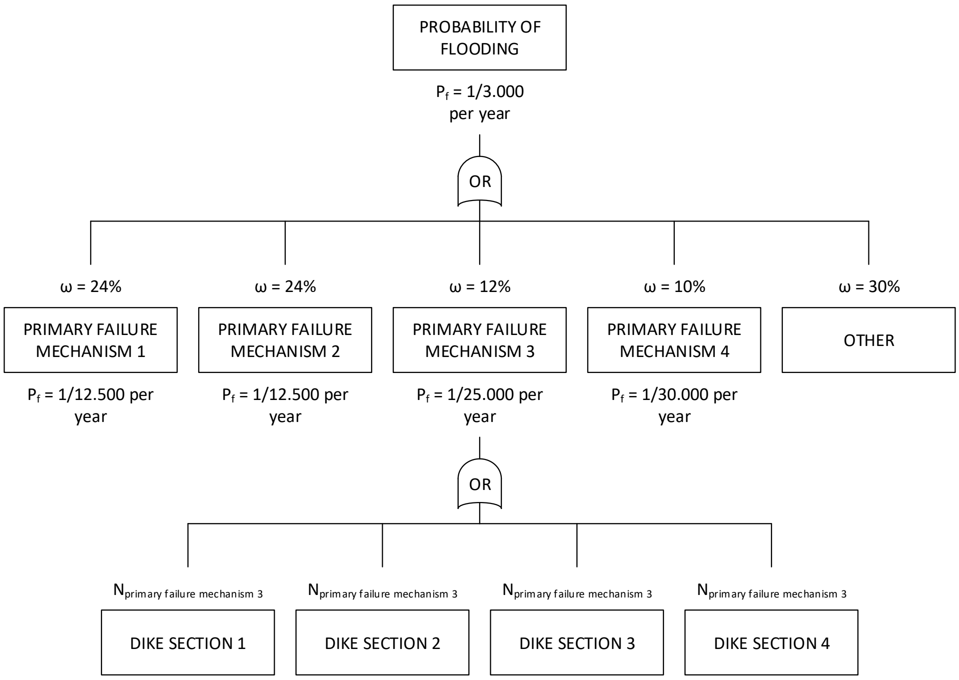


Figure 3.11: Assigning technical requirements, given a fictitious failure budget, for a probability of failure ($P_f$) of 1/3.000 per year.

## 3.3. The Length-Effect

A common used analogy to explain the phenomenon discussed in this section is the example of a loaded chain. The longer the chain, the more links included and thus the larger the uncertainty in the strength parameters, resulting in a higher probability to contain a weak spot.[9] This works similar for flood defences. The increase of the probability of failure with respect to the total length of the dike segment is referred to as the length-effect. As explained, a dike segment consists of multiple contiguous components. Since components differ in size and are constructed of different materials, their contribution to the length-effect varies. The length of dike and dam sections vary between 150 m to near 2 km. In combination with large spatial uncertainties in some of the resistance parameters due to low spatial data resolution of subsoil information, the technical requirements associated to these components are most sensitive to the length-effect. In combination with the scope of this research, in this section only the length-effect affecting dike and dam sections is conceptually discussed. Firstly, in subsection 3.3.1 the method to account for the length-effect in translating the safety standard in cross-sectional reliability requirements is explained. Secondly in section 3.3.2 the factors that determine the magnitude of the length-effect factors are discussed. Lastly, in section 3.3.3 the simplified method and the associated default values used to determine the magnitude of the length-effect factors for the primary failure mechanisms piping and slope stability are shown.

### 3.3.1. Accounting for the Length-Effect

As explained in the previous subsection, the length-effect factors, $N$, are used to transform the permissible probability of failure per failure mechanism for the dike segment into technical requirements of a representative cross-section per component. Equation 3.1 is used to conduct this transformation.

$$P_{\textit{technical requirement i}} = \frac{P_{\textit{dike segment}} \cdot \omega_{\textit{primary failure mechanism i}}}{N_{\textit{primary failure mechanism i}}} \tag{3.1}$$

where:

| | | | |
|---|---|---|---|
| $P_{\textit{technical requirement i}}$ | = | Cross-sectional probability of failure of primary failure mechanism i | [$y^{-1}$] |
| $P_{\textit{dike segment}}$ | = | Probability of flooding of the dike segment | [$y^{-1}$] |
| $\omega_{\textit{primary failure mechanism i}}$ | = | Failure probability budget factor for a primary failure mechanism i | [-] |
| $N_{\textit{primary failure mechanism i}}$ | = | Length-effect factor of primary failure mechanism i | [-] |

### 3.3.2. The Magnitude of the Length-Effect

Since each primary failure mechanism depends on specific load and resistance parameters, the magnitude of the length-effect varies per primary failure mechanism. Therefore, the length-effect factors are determined independently per primary failure mechanism. Explained in subsection 3.2.3, the dike segment contains of contiguous series of components, chosen such that per component stochastic homogeneity of both the load and the resistance is obtained. However, not all components are equally prone to each primary failure mechanism and thus vary the magnitudes of the length-effect factors between the primary failure mechanisms. To quantify the magnitude of the length-effect factors, at first the ratio of the dike segment length (the accumulative length of the sensitive components) that contributes to the length-effect per primary failure mechanism has to be determined. Secondly, the question is to which extent the failure probability increases with length [25]. Due to spatial correlations of both the load and resistance parameters, the procedure of quantifying this extent has to involve both the dependency between the components and the extrapolation of cross-section reliability to component reliability [26].

Vrijling et al. [50] concludes that mainly the following two factors determine the magnitude of the length effect:

1. The relative contribution of the associated load and resistance with respect to the total variance. A large contribution of the resistance parameters increases the magnitude of the length-effect, because mostly the fluctuation of load parameters of flood defences, e.g., the water level, are of a larger scale than the resistance parameters.

2. The spatial variability of the subsoil. The magnitude of the length-effect increases in case spatial variation increases, since the probability of containing a weak spot increases.

[9] A weak spot is an unfavourable realisation of a load and strength parameter combination or a local anomaly.

### 3.3.3. Default Values

In practice, the data for the probabilistic analysis to derive length-effect factors as explained in the previous subsection is often missing, meaning that decisions have to be made based on the available material and engineering judgment [24]. This led to different simplified derivation methods. For slope stability and piping, the length-effect factors are calculated according to Equation 3.2.

$$N_{slope\ instability,\ piping} = 1 + \frac{a \cdot L_{dike\ segment}}{b} \tag{3.2}$$

where:

| | | | |
|---|---|---|---|
| $a$ | = | Part of the dike segment sensitive to the primary failure mechanism | [-] |
| $L_{dike\ segment}$ | = | Dike segment length | [m] |
| $b$ | = | Independent equivalent primary failure mechanism length | [m] |

The formulas for both primary failure mechanisms are accompanied with default values have been determined, which has been calibrated to a reference database for the Netherlands. Identical to the standard failure probability budget factors, depending on the local situation, deviation from these values is allowed.[10] In Table 3.1 the standard values for $a$ and $b$ are presented. More information regarding the origin of these values can be found in Jongejan [24].

Table 3.1: $a$ and $b$ values for the primary failure mechanisms slope stability and piping.

| **Primary failure mechanism** | **Region** | ***a*** | ***b*** |
|---|---|---|---|
| Slope Stability | All | 0.033 | 50 |
| Piping | Upper Rivers | 0.90 | 300 |
| | Other | 0.40 | 300 |

The contribution of soil stratification uncertainties is incorporated in the estimation of parameter $a$, which magnitude is based on expert judgment given the local circumstances per dike segment. In section 8.2 it is explained how the insights in soil stratification that are derived by the proposed Machine Learning method can lead to an update or a strengthening of the estimation.

[10] According to Jonkman et al. [25], the application of default values for especially $a$ and to some extent $b$ can lead to conservative estimates of the actual reliability or required dimension to meet the safety standard. They mention that the chosen $a$ values for the primary failure mechanism piping can be lowered considerably in the upper river area through a geotechnical site investigation.

# 4
# DEFORMATION OF THE SUBSOIL

When a load is applied on top of a compressible subsoil, volume change takes place through relocation of the soil particles, or sometimes due to particle fracture. In both cases the volume of the soil particles remains constant, which indicates that the void volume decreases. This means that either air or water is being squeezed out of the soil. Over time, the soil starts to deform, which leads to measurable settlement of the surface. In section 4.1 the theory behind deformation is explained and in section 4.2 is discussed how the deformation theory can be modelled and which is computer program is used to conduct the calculations.

The information provided in this chapter is mainly based on *Introduction to Soil Physics* by Hillel [21], *Principles of Geotechnical Engineering* by Das and Sobhan [16] and *Soil Mechanics* by Verruijt [49].

## 4.1. Theory of Deformation

In this section the physical processes behind soil deformation are discussed. According to Hillel [21], the variation between the viscosity of air and water results in a significant difference in the time scale of the deformation process in dry and saturated soils. Das and Sobhan [16] describe three processes that can be distinguished: initial compression, primary consolidation and secondary consolidation, which are discussed in respectively subsection 4.1.1, 4.1.2 and 4.1.3. In Figure 4.1 the distinction between the stages of a settlement curve is illustrated.

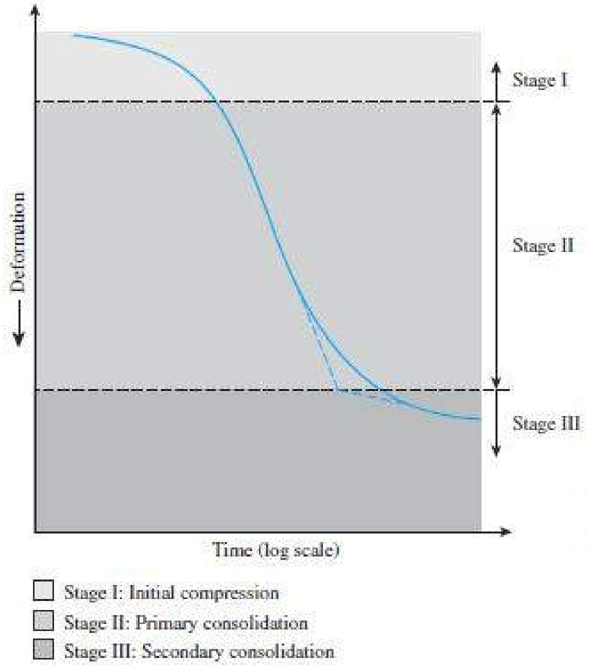


Figure 4.1: The three distinguishable deformation stages of a settlement curve: initial compression, primary consolidation and secondary consolidation. Reprinted from *Principles of Geotechnical Engineering*, by Das, 2012.

### 4.1.1. Initial Compression

When dry or partially saturated compressible soils are being loaded, initial compression occurs immediately after the load is applied. In order to support the new load, the soil matrix is densified. This densification process is described by the theory of elasticity, wherein the soil volume decreases elastically due to expulsion and compression of air in the voids without a change in the moisture content. Due to the low viscosity of air, the time scale of initial compression is in the order of days. In saturated compressible soils initial compression is limited or not present, since the pore water pressure support the load immediately after it is applied. Compared to the magnitude of the associated process, called primary consolidation, that occurs in such case, the magnitude of initial compression is several times smaller or even negligible. In subsection 4.1.2 primary consolidation is discussed more in detail.

### 4.1.2. Primary Consolidation

In saturated compressible soils, after the expulsion of air, the initial compression stage is followed up by the primary consolidation stage, which was firstly described in Terzaghi [44]. In contrast to initial compression, as explained in subsection 4.1.1, instead of air, water has to be squeezed out in order to densify the soil matrix. However, due to its high viscosity with respect to the viscosity of air, this process is not an instantaneous response of the soil. The result is that instead of immediate densification of the soil matrix, the pore water pressure supports the load increment. Over time, the excess pore water pressure will dissipate and the soil matrix start to support the load through densification. This process, called primary consolidation, exists of a linear elastic and a non-linear elastic response. The distinction between these responses is described by the spring analogy, shown in Figure 4.2.

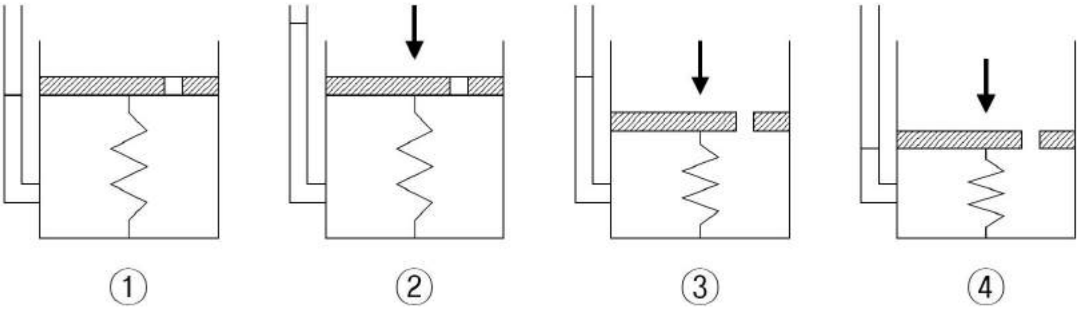


Figure 4.2: The spring analogy used to describe primary consolidation. Reprinted from *Wikipedia*, n.d., Retrieved December 5, 2018, from `https://en.wikipedia.org/wiki/Consolidation_(soil)`.

Plot 1 in Figure 4.2 shows a reservoir filled with water, wherein a plate attached to a spring floats on the water surface. In this scenario the spring represents the soil matrix and the water the filled voids between the soil particles. Since the spring carries the weight of the plate, the excess pore water pressure (the hydraulic gradient between the water levels in the tube and the reservoir) is equal to zero, which indicates there is equilibrium between all forces. In plot 2, a load is applied of top of the plate, resulting in distortion of the equilibrium. The gap in the plate allows the water to flow away, therefore forcing the spring to support the newly applied load (linear elastic response). However, the gap in plate is not large enough to let the water flow away immediately (non-linear elastic response). The result is a temporary increase of the excess pore water level, as shown in the tube. Over time, the water dissipates through the gap and the excess pore water pressure reduces (plot 3), until a new equilibrium has been found (excess pore water pressure = 0) wherein the spring has been compressed such that it is capable of supporting the new load.

This analogy represents precisely the process that occurs during the primary consolidation stage. To stiffness of the spring is equal to the coefficient of volume compressibility ($m_v$) and velocity of the water flowing to the gap is equal to the coefficient of permeability ($k$). Given the magnitude of the load and these two parameters, the settlement rate of the plate can be determined. Often the settlement rate is directly expressed by the coefficient of consolidation, which is denoted by $c_v$. The relation between $m_v$ and $k$ is shown by Equation 4.1.

$$c_v \quad = \quad \frac{k}{\gamma_w \cdot m_v} \tag{4.1}$$

where:

| | | | |
|---|---|---|---|
| $c_{\mathrm{v}}$ | = | Coefficient of consolidation | [m$^2$/s] |
| $y_{\mathrm{w}}$ | = | Unit weight of water | [kN/m$^3$] |
| $k$ | = | Coefficient of permeability | [m/s] |
| $m_{\mathrm{v}}$ | = | Coefficient of volume compressibility | [m$^2$/kN] |

### 4.1.3. Secondary Consolidation

In clay and peat layers, the primary consolidation stage is followed up by the process called secondary consolidation. The process is firstly reported circa 1932 by Keverling Buisman, who named it the secular effect [49]. After the excess pore water pressure has been dissipated during the primary consolidation stage, as described in subsection 4.1.2, the reduction in volume continues at a very slow rate. In accordance to Verruijt [49], Barden [6] concludes that secondary consolidation of clays is caused by plastic deformation of soil particles under a constant stress (creep) and viscous behaviour of the clay-water system, but yet no agreement is found that describes the physical processes that cause secondary consolidation of peats.

## 4.2. Modelling of Deformation

In order to model one-dimensional compression of a soil column, several models have been developed. In den Haan [19] it is stated that many of those models are derived based on incorrect assumptions, since it was believed that the primary and secondary consolidation processes were consecutive phenomena. Finally, through accumulation of knowledge over time, the isotache model was developed. In this model both components are separate constitutive effects. Nowadays, the isotache model is seen as the state-of-the-art model to simulate one-dimensional compression. Verruijt [49] mentions that, although internationally slightly different formulas with different constants are being used (NEN-Bjerrum versus a,b,c-isotache), there is a general agreement on the basic form of the relations (with a logarithm in time), as derived in den Haan [18]. Both formulas offer equivalent results in case of small deformations, but in case of large deformations the a,b,c-isotache model becomes advantageous as it is based on natural strain and it uses a rate type formulation (inelastic compression is assumed to result from visco-plastic creep) [17]. Since the WBI-SOS subsoil scenarios discussed in subsection 2.3.2 contain peat and clay layers, large deformations are expected. Therefore, in this research the synthetic surface settlement curves will be calculated by means of the a,b,c-isotache model.

### 4.2.1. a,b,c-isotache Model

In this subsection the solution of the a,b,c-isotache model is briefly explained.[1] The model obtains the natural strain by defining the increment of strain relative to the present, actual layer thickness, and by integrating the increments according to Equation 4.2 [17].

$$\epsilon^H \quad = \quad -\int_{h_0}^{h} \frac{\mathrm{d}h}{h} \quad = \quad -ln\left(\frac{h}{h_0}\right) \tag{4.2}$$

where:

| | | | |
|---|---|---|---|
| $h$ | = | Actual layer thickness | [m] |
| $h_0$ | = | Initial layer thickness | [m] |

The solution to obtain the magnitude of natural strain for a soil layer during a constant stress period after the new load has been applied is presented by Equation 4.3.

[1] For more information regarding the a,b,c-isotache model, it is referred to *"Vertical Compression of Soils"* by den Haan [18].

$$\epsilon^H(t) \quad = \quad a \cdot ln\left(\frac{\sigma_p}{\sigma'_0}\right) + b \cdot ln\left(\frac{\sigma'}{\sigma_p}\right) + c \cdot ln\left(\frac{\tau}{\tau_0}\right) \tag{4.3}$$

where:

$$\tau \quad = \quad t + t_{age} \tag{4.4}$$

with the initial equivalent age:

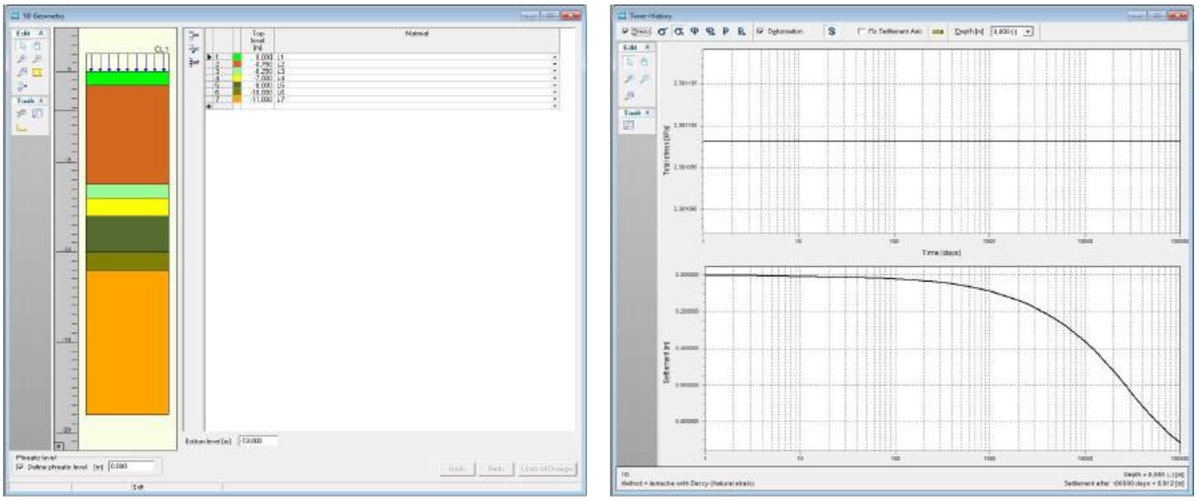
$$t_{age} \quad = \quad \tau_0 \cdot \mathrm{OCR}^{\frac{b-a}{c}} \tag{4.5}$$

where:

| | | | |
|---|---|---|---|
| $\epsilon$ | = | Natural vertical strain (Hencky) | [-] |
| $a$ | = | Modified natural swelling index | [-] |
| $b$ | = | Modified natural compression index | [-] |
| $c$ | = | Modified natural secondary compression constant | [-] |
| $\sigma_p$ | = | Preconsolidation pressure | [kN/m$^2$] |
| $\sigma'$ | = | Effective vertical pressure | [kN/m$^2$] |
| $\sigma'_p$ | = | Initial effective vertical pressure | [kN/m$^2$] |
| $\tau_0$ | = | Reference time, set to 1 | [days] |
| $t$ | = | Time | [days] |
| $OCR$ | = | Overconsolidation Ratio | [-] |

### 4.2.2. D-Settlement

In order to conduct calculations by the a,b,c-isotache model, the computer program D-Settlement is used. In this subsection the required input and the output of the program is briefly explained.[2] In D-Settlement a soil column in terms of WBI-SOS units is imported and the load of the new soil layers is placed on top of the column. Subsequently, the initial conditions are correctly set. Preloading is accounted for by the OCR value and the preconsolidation pressure ($\sigma_p$) per soil layer. In case an initial creep velocity caused by the load of the old dike has to be accounted, an initial load corresponding to the local equivalent load of the old dike body is set on top of the soil column. The ground water table is set to match the preferable saturation rate. Since the subsoil is assumed to be fully saturated at all times, the water table is set equal to the top level of the soil column (-1.0 m NAP). Finally, the program calculates the vertical settlement of the column over a time span 100 000 days. Figure 4.3 shows an example of the input (Figure 4.3a) and the associated output (Figure 4.3b).



(a) D-Settlement input: the imported soil column composed of WBI-SOS units with the correct load and initial conditions set.

(b) D-Settlement output: calculated settlement curve on log scale (bottom plot) given a certain loading curve (top plot).

Figure 4.3: The D-Settlement environment.

[2] For more information regarding the numerical solution of the a,b,c-isotache model and the corresponding assumptions is referred to *"D-Settlement - User Manual"* by Deltares [17].

# 5

# MACHINE LEARNING

In this chapter the concept of Machine Learning is explained. In section 5.1 the concept of Machine Learning and the parameters which can be used to navigate through the wide variety of existing paradigms within the field of Machine Learning are explained. Secondly, in section 5.1.2 the best matching paradigm for the learning problem as described in subsection 2.1 is chosen. Section 5.3 elaborates the general theory regarding Supervised Learning and finally in section 5.4 the algorithm that will be used in this thesis is chosen and its functionality is explained.[1]

The information in this chapter is mainly based on *Understanding Machine Learning* by Ben-David and Shalev-Shwartz [8], *Learning From Data* by Abu-Mostafa et al. [3] and *Efficient Learning Machines - Theories, Concepts, and Applications for Engineerings and System Designers* by Awag and Khanna [5].

## 5.1. Brief Introduction to Machine Learning

This section provides an introduction of the concept of Machine Learning and the parameters which can be used to navigate through the wide variety of existing paradigms within the field of Machine Learning is given. In conclusion of this section, the best matching paradigm to the learning problem as described in this thesis is chosen.

### 5.1.1. What is Machine Learning?

Recently, Machine Learning has gained fresh momentum due to the explosion in the amount of accessible data in the world, often called *Big Data*, rapidly growing computational capacity and affordable data storage. Machine Learning offers the opportunity to solve highly complex problems, that are either impractical to solve each time by inventing specialised algorithms or impossible to solve at all. An oft-quoted statement to describe this scientific field is made by Samuel [39] in 1959:

> *"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed."*

The term Machine Learning stands for automated learning. The idea of automated learning is that a computer program is able to “learn” from input made available to them. Mitchell [32], who converted this idea of learning by a computer program into a more concrete perspective, devised the following definition:

> *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*

He explained that the input of the computer program is the data, representing the experience E. Running this through a computer program, the output is the skill or knowledge, defined by task T. The quality of the learning process that took place is determined by the rate of success in performing task T, visualised by the performance measure P.

[1] Although usually no conclusion are made in the literature review, due to the enormous wide variety within the field of Machine Learning, in this case an exception is made. With the decision to conclude already in this chapter the best matching paradigm and the algorithm that will be used, it becomes possible to narrow the associated sections to what is of relevance for this thesis, increasing the readability and comprehensibility of the thesis.

## 5.1.2. A Guideline Through the Field of Machine Learning

According to Ben-David and Shalev-Shwartz [8], learning itself is defined as the acquisition of skills or knowledge through study, experience or being taught. The acquisition of skills or knowledge through a certain action is called a paradigm. Depending on the situation, one learns through one of these paradigms. Since Machine Learning has to deal with various situations, the field has been subcategorised into a manifold of paradigms. Each paradigm is specialised in dealing with a different type of situation, also called the learning problem. Ben-David and Shalev-Shwartz [8] describes four parameters along which the variety of paradigms can be classified:

- Supervised Learning versus Unsupervised Learning versus Reinforcement Learning
- Active Learning versus Passive Learning
- Online Learning versus Batch Learning
- Helpfulness of the Teacher

To get an understanding of these parameters, typically the following roles and spaces are defined:

Teacher = The user.
Learner = The Machine Learning algorithm.
Input Space = Data domain on which the acquired skill through learning can be applied and from which the knowledge is derived.
Training Set = Sample originating from the input space on which the skill is trained.
Validation Set = Sample originating from the input space on which the skill is validated.

### Supervised Learning versus Unsupervised Learning versus Reinforcement Learning

The first parameter describes the nature of the interaction between the learner and the input space. Three paradigms can be distinguished: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

One speaks of Supervised Learning in case the learning process is performed on a sample from the input space, chosen by the teacher. This sample is called a training set. This training set contains inputs and corresponding outputs, called labels. Based on this training set the learner is taught how to act in the environment. The performance of the learning process will be tested on a validation set. It can be seen as if the learning process is being “supervised” by the teacher before the learner will act in the complete input space. The size of the training and validation set should be chosen such that it forms a representative of the input space. Nonetheless, one should take into account, once the learning process uses labelled information, the output is always biased with respect to the chosen training set.

In case the learning process is directly performed on the input space, one speaks of Unsupervised Learning. The learner has the task to process the complete input space and provide information about it. No training or validation set is used during the learning process. The main advantage is the non-biased outcome. The difficult part is the size of the input data and the validation of the outcome. First of all, the input space in reality is often wide. Hence, the size of the data input becomes extremely large. Second, since there are no labels, the output is not defined. Validating the quality of the learning process becomes therefore more difficult.

When the learner can only observe a part of the input space at a time, the setting is called Reinforcement Learning. Based on this observed state, the learner has to take actions in order to receive rewards, with respect to reaching a certain goal. The input space processes the action and reveals the new state to the learner including a reward, related to the performance of the chosen action. Given this feedback, the learner decides the next action. This iterative process continues until the goal is reached. During each iteration step, the amount of decisions that can be made are based on the degree of freedom within the input space, determined by the teacher. The task of the learner is to rank the possible actions and choose the one with the highest estimated cumulative reward. The cumulative reward presents for each action the long-term reward, obtained after reaching the goal by pursuing a particular strategy, discounted into a short-term reward regarding the state of the input space. The learning process can be seen as the creation of a training set that includes all possible actions and the associated cumulative rewards, for any strategy and any state of the input space. This learning process can be performed prior to the exposure to the input space, in case the decision-making process demands speed. However, depending on the degree of freedom within the input space, the size of the training set could become significantly. In case memory capacity is limited, one can also choose to make the learning process an adaptive process to the presented states of the input space. This, in turn, will negatively affect the speed of the decision-making process.

#### Active Learning versus Passive Learning

The second parameter describes the role played by the learner during the learning process. A distinction can be made between Active Learning and Passive Learning.

An Active Learner seeks for contact with the teacher during the learning process in order to enhance its accuracy. This could be by proposing suggestions or performing experiments. In contrast, a Passive Learner does not seek for any contact and goes through the learning process without influencing or directing training.

#### Online Learning versus Batch Learning

The third parameter describes the nature of the training set data during the learning process. A distinction can be made between Online Learning and Batch Learning.

With Online Learning the learner uses data that is presented by a stream during the learning process. The amount of data will increase over time. At the start of the learning process the performance is often poor, however it will become more trustworthy as time goes by. With Batch Learning the learner has accessibility to large amounts of data during the learning process. During the learning process the application can be fine-tuned to meet preferable outcomes. The performance measured at the end of the learning process indicates the life time performance.

#### Helpfulness of the Teacher

The last parameter describes the impact on the learning process by the teacher. The role the teacher plays in the learning process could be distinguished in being passive or active.

The passive teacher only communicates information from the input space to the learner. The learner postulates this information as being generated by a random process. This means all information is of equal importance to the learner. In case the teacher plays an active role, the learner gets help during the learning process. The teacher selects the most useful information in order to stimulate the learning process. Even though the selection process causes a biased outcome, it often enhances the performance of the learning process nonetheless. It could also be that an active teacher tries to mislead the learner, by selecting incorrect information or trying to manipulate the learning process in a negative way. The advantage of learning against an adversarial teacher is obtaining knowledge about the robustness of the Machine Learning application. In reality it is often tried to undermine systems. Trying to mislead the learner already during the learning process provides an application that is resistance against known threats.

## 5.2. Assign a Paradigm to the Learning Problem

In this section a paradigm is assigned to the learning problem as explained in section 2.1 by using the parameters that are described in subsection 5.1.2.

In general the quantity of subsoil measurements within engineering projects is in the order of tens to maximum a couple of hundreds. Since an Unsupervised Learning approach requires millions of data points, with the current available subsoil measurement techniques it is yet not reasonable to expect this might be an option. Because the complete input space from the three-dimensional surface settlement monitoring data and the associated load curve is already known prior to the learning process, Reinforcement Learning does not match the learning problem.

Both Online and Batch Learning could be used for the learning problem as described in thesis. Because only a proof of concept is given in this thesis, out of simplicity the question how the prediction accuracy relates to the length of the time series of the synthetic surface settlement curves are neglected, hence Batch Learning is applied.

The learning problem as described in this thesis: obtaining insights in the spatial variation of soil stratification, is of a physical nature. In order to obtain an independent understanding of such learning problems, the accessible information from measurements and observations should be managed in an unbiased way. Therefore, it is chosen to use a Passive Learner taught by a Passive Teacher.

In conclusion, in this research the possibility of a Supervised Learning approach, with a Passive Learner which learns from Batched training data and is taught by a Passive Teacher is investigated. A similar approach has been presented in Osisanwo et al. [34] to classify diabetes test results. This project shares the physical nature of its learning problem and aims to obtain an unbiased solution.

## 5.3. Supervised Learning Theory

In this section general Supervised Learning theory is discussed. First, the components of Supervised Learning are shown and secondly the principle of generalisation is explained. Thirdly, two theories to determine the out of sample error are discussed: the Vapnik-Chervonenkis Analysis and the Bias-Variance Analysis. Fourthly, the concepts of underfitting and overfitting are elaborated and lastly the tuning possibilities to decrease the out of sample error are shown.

### 5.3.1. Supervised Learning Components

The learning process associated to Supervised Learning contains in general the direct and indirect components, which are summarised and explained in the list below. Although some of the components have already been described in the previous chapters and sections, to complete the summary they are described twice.

Feature Vector = Collection of features, that together characterise the object that is being analysed.
Label = Representation of object that is being described by the feature vector.
Training Data = Collection of labelled feature vectors
Input Space = Data domain on which the acquired skill through learning can be applied and from which the knowledge is derived.
Output Space = Prediction results of the input space.
Target Function = The unknown function that is theoretically capable of transforming the input space in the preferred output space with an accuracy of 100%.
Hypothesis = Learning algorithm of a certain complexity, which is expressed in the number of free parameters.
Hypothesis Set = Collection of hypothesis that will be tested.
Error Measure = Function that estimates the prediction error of the hypothesis in predicting the output space.
Final Hypothesis = The hypothesis from the hypothesis set that approximates most accurately the target function.
Learning Model = Fixed training set-up in which each hypothesis part of the hypothesis set is trained on the training set and the associated prediction accuracy is estimated by the error measure.

To maintain consistency, through the rest of this thesis the following related formalisation of the components will be used:

| | | |
|---|---|---|
| Feature Vector | = | $x$ |
| Label | = | $y$ |
| Training Data | = | $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ |
| Input Space | = | $X$ |
| Output Space | = | $Y$ |
| Target Function | = | $f : X \rightarrow Y$ |
| Hypothesis | = | $g : X \rightarrow Y$ |
| Hypothesis Set | = | $H$ where $g \in H$ |
| Error Measure | = | $E(g)$ |
| Final Hypothesis | = | $g + \min(E(g)) \approx f$ or $P(y \mid x)$ |

In Figure 5.1 the connections between the components are shown in the learning diagram.

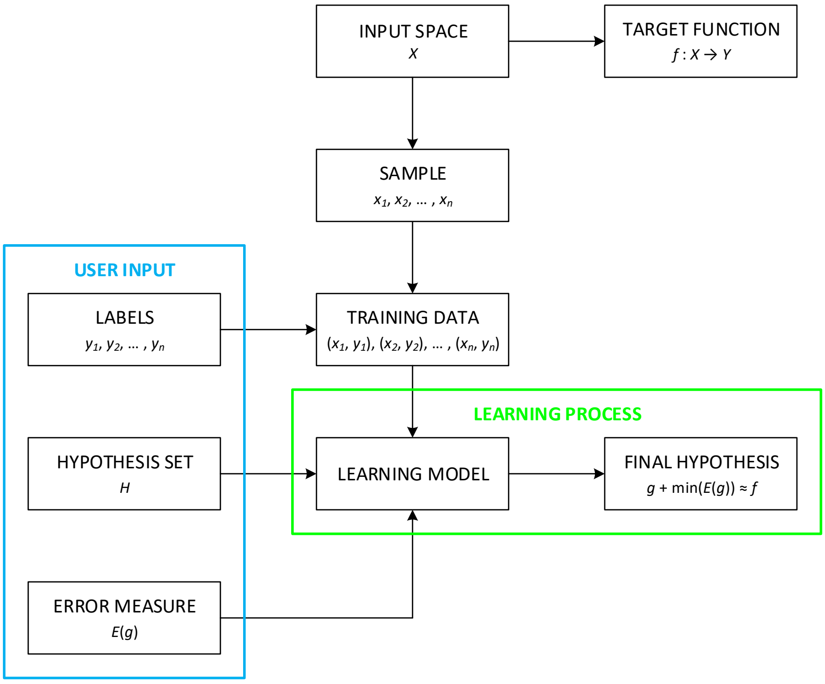


Figure 5.1: The learning diagram.

### 5.3.2. Principle of Generalisation

The principle of generalisation is the key component within Supervised Learning. It refers to the ability of the trained algorithm to provide accurate predictions on the input space, rather than just the training data it was trained on. Because the algorithm will autonomously predict the output space after it is being trained, it is important to know the expected quality of the predictions. Unfortunately, this prediction accuracy can only be estimated during a correctly conducted learning process.[2] Within a correctly conducted learning process the difference in the quality of the prediction accuracies of the target function and the final hypothesis is quantified by the measured in sample error, denoted by $E_{in}(g)$, which guarantees to approximate the unknown out of sample error, denoted by $E_{out}(g)$. Finally, if small $E_{in}(g)$ values are derived and generalisation is ensured, it can be stated that the algorithm performs well out of sample. Hence, it is likely that learning took place. Given the $E_{out}(g)$ and the prediction accuracy measured during the training process, the expected prediction accuracy in predicting the output space can be determined.

The following situations do negatively affect generalisation, resulting in either underfitting or overfitting. These phenomena and associated prevention techniques are individually discussed in subsection 5.3.4.

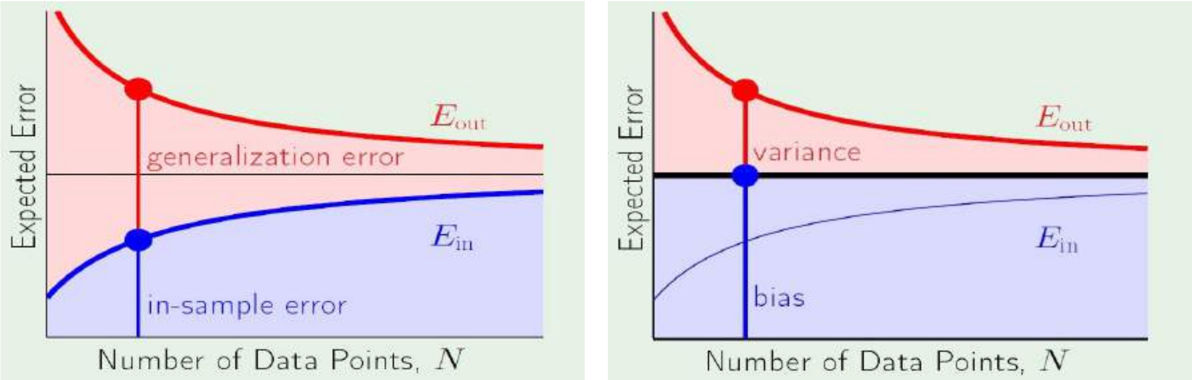
- The quantity of training data is not sufficient to present the complexity of the input space (underfitting).
- The complexity of the hypothesis set is not matching the complexity of the training data (underfitting/overfitting).
- Manipulation of the learning process by using human intelligence (overfitting).

For all situations the quantity of the training data influences the magnitude of underfitting and overfitting. Therefore, learning curves are used to discuss the concepts in the subsequent subsections, which show the expected $E_{out}(g)$ and the measured $E_{in}(g)$ as a function of the quantity of the training data for a certain hypothesis.

[2] How to conduct a correct learning process is explained in subsection 5.3.5.

### 5.3.3. Estimating the Out of Sample Error

In order to estimate the magnitude of the out of sample error, $E_{out}(g)$, several methods have been developed. According to Abu-Mostafa et al. [3], the two most well-known methods are the Vapnik-Chervonenkis Analysis and the Bias-Variance Analysis. In Figure 5.2 a comparison of learning curves corresponding to both techniques is shown.



(a) The Vapnik-Chervonenkis Analysis.

(b) The Bias-Variance Analysis.

Figure 5.2: Visualisation of two techniques to estimate the generalisation error given a fixed hypothesis set: the Vapnik-Chervonenkis Analysis and the Bias-Variance Analysis. Reprinted from *Learning From Data*, by Abu-Mostafa, 2012.

Contradictory to the components of the expected error as presented in both Figure 5.2a and Figure 5.2b, in reality also the irreducible error is part of the total $E_{out}(g)$. This irreducible error is caused by stochastic noise of the input space by measurements uncertainties, etc. In this thesis, given the made assumptions that synthetic surface settlement measurements do directly represent the subsoil settlement and measurement uncertainties are negligible, the input space does barely contain any stochastic noise, therefore the influence of stochastic noise is excluded from the following explanations.

#### Vapnik-Chervonenkis Analysis

The Vapnik-Chervonenkis Analysis (VC Analysis) estimates $E_{out}(g)$ by binding it to the measured $E_{in}(g)$ with the so called generalisation bound. The generalisation bound is a function to describe magnitude of the generalisation error, the difference between the expected $E_{out}(g)$ and the measured $E_{in}(g)$, given the Vapnik-Chervonenkis Dimension (VC Dimension) and the quantity of the training data. The VC Dimension present the maximum number of times the hypothesis set can be shattered, given the quantity of the training data.[3] In case the complexity of the hypothesis set is related to a finite VC Dimension ($d_{VC}(H)$), one can proof that the final hypothesis ($g \in H$) will generalise. In Figure 5.2a it can be seen that an increasing quantity of training data decreases the generalisation error, given a fixed hypothesis set. In case the composition of the hypothesis set exceeds the complexity dictated by the VC Dimension, the generalisation bound is violated and therefore the guarantee that the generalisation error remains small disappears.

#### Bias-Variance Analysis

As the name suggests, the Bias-Variance Analysis estimates $E_{out}(g)$ from a composition of the error corresponding to the Bias and the Variance of a certain learning model. The Bias is the error that defines the structural difference between the final hypothesis and the unknown target function, due to assumptions that are corresponding to the algorithm. The Variance is the variable error of the final hypothesis for any given prediction, due to learning a single data set. Both the Bias and the Variance are calculated through Validation, which is explained in subsection 5.3.5. The Bias and Variance are inherently coupled to each other. Given a certain quantity of training data, in order to decrease the Bias, one has to increase the complexity of the learning model. At the same time, a more complex learning model learns the specific training data better, resulting in an increase of the Variance. This inevitable trade-off is called the Bias-Variance Trade-off. In Figure 5.2b the optimal trade-off is shown as function of the quantity of training data, given a fixed hypothesis set.[4] Due to using a fixed hypothesis set, the bias associated to the final hypothesis (black bar) remains constant over

[3] More detailed information about the Vapnik-Chervonenkis Dimension and the corresponding mathematical derivation, can be found in *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities* by Vapnik and Chervonenkis [46].

[4] How to find the optimum of the Bias-Variance Trade-off by means of the Bias-Variance Analysis is explained in subsection 5.3.5.

an increasing quantity of training data. In contrast, the probability of the learning model to learn the training data decreases, therefore the Variance is decreasing. In case the composition of the hypothesis set changes, a new optimum of the trade-off will accordingly shift the black bar in the vertical direction and adjust the position and the slope of the $E_{out}(g)$ curve.

### 5.3.4. Underfitting and Overfitting

The situation which are related to underfitting and overfitting are already briefly mentioned in subsection 5.3.2. In this subsection the causes and the concepts themselves will be elaborated. Figure 5.3 illustrates the concepts with respect to an appropriate fit.

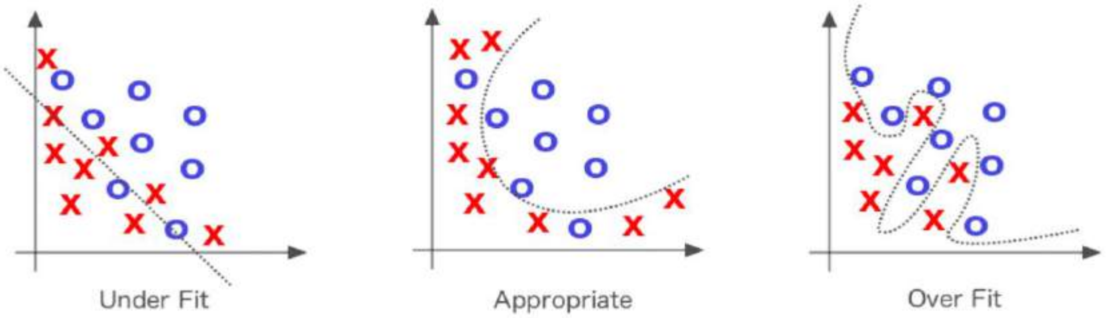


Figure 5.3: The concept of underfitting and overfitting a classification problem with respect to an appropriate fit. Reprinted from *Deep Learning*, by Patterson and Gibson, 2017.

#### Underfitting

With respect to the Bias-Variance Trade-off, underfitting results in a high Bias. In case both $E_{out}(g)$ and $E_{in}(g)$ are high, one knows that the prediction accuracy is subjected to underfitting. The cause could either be a lack of training data to represent the complexity of the input space or a lack of complexity of the hypothesis set. The first cause is a difficulty that often occurs if the quantity of the training data is low with the respect to the complexity of the input space. In this case the quantity of the training data has to increase and/or the complexity of the input space has to be reduced. The second cause is not often the case, however in case it is, it is easily to solve by composing a hypothesis set of more complex algorithms.

#### Overfitting

With respect to the Bias-Variance Trade-off, overfitting results in a high Variance. In case $E_{out}(g)$ is high but $E_{in}(g)$ is low, one knows that the prediction accuracy is subjected to overfitting. It is more harmful compared the underfitting, since it results in a misleading "in sample" error which promises a higher prediction accuracy then the algorithm actually can provide. The causes resulting in overfitting can be of different natures.

First, the complexity of the composed hypothesis set exceeds the complexity of the training data. The basically means that the algorithm is learning the stochastic noise instead of a pattern. An increase of the quantity of training data will decreases the magnitude of overfitting. Alternatively, the complexity of the input space could be expressed by more features. In case of a fixed quantity of training data, which is mostly the case, the complexity of the hypothesis set has to be decreased through composition of less complex algorithms.

The second cause is manipulation by using human intelligence in the learning process. Some paradigms use manipulation to increase the robustness of the algorithm, however the idea of a Passive Learner whom is taught by a Passive Teacher is to learn purely given the provided training data. In case human intelligence is interfering in such learning process, the result might be that an algorithm learns incorrectly and therefore does not generalise. Manipulation can be intentional or unintentional:

- Intentional manipulation is for example using the human intelligence to find a pattern in the training set prior to the training phase and steer the learning model already in the "good" direction by accordingly composing the hypothesis set. This limits the neutrality of the the Passive Learner to learn the patterns presented by the training data and increases the probability of learning incorrect patterns which do not generalise. This type of manipulation is often applied in case of a small quantity of training data. It has to be stated that intentional manipulation might lead to a better prediction accuracy in some situations, however, one has to accept the risk that goes along with violation the principle of generalisation.

- Unintentional manipulation is more dangerous, since one is not aware has occurred. It often occurs due to small mistakes made prior and during learning process. An example of unintentionally manipulation prior to the learning process is data scraping of all the training data, instead of only the training and validation set.[5] Data scraping is conducted in order to remove stochastic noise, according to the observations of human intelligence. The idea is that after data scraping the features are "clean" and therefore the quality of the learning process will increase. However, the risk of data scraping is that certain characteristics of the input space are accidentally removed. In case all the training data is scraped, also the test set contains human knowledge due to the removal. Therefore, it is polluted and not capable to function as reference anymore, as testing the performance on the polluted test set troubles the error measures. Consequently, the guarantee that $E_{in}(g)$ mirrors $E_{out}(g)$ disappears. In case one applies data scraping, this can be prevented by directly putting the test set apart after splitting the training data and make sure it remains in its original shape.

### 5.3.5. Tuning Possibilities

In case the VC Analysis to estimate $E_{out}(g)$, one has to assure the complexity of the chosen hypothesis set does not exceed the VC Dimension, which is determined by the quantity of the training data. In case of few quantities of training data, this approach limits the complexity of the hypothesis set that can be tested and therefore lowers the probability of finding the best algorithm to solve the learning problem. In contrast, using the Bias-Variance Analysis in combination with cross-validation offers the opportunity of tuning the learning model, which means a complex hypothesis can be used without violating the principle of generalisation. This increases the probability of finding the best algorithm to solve the learning problem.

During tuning of the learning model one chooses the final hypothesis that provides the smallest combination of the Bias and the Variance, resulting in the smallest estimation of $E_{out}(g)$. Figure 5.4 visualises how the final hypothesis is found from the optimum of the Bias-Variance Trade-off. Hypotheses that have a lower complexity are subjected to underfitting, while the ones that are of a higher complexity are subjected to overfitting.

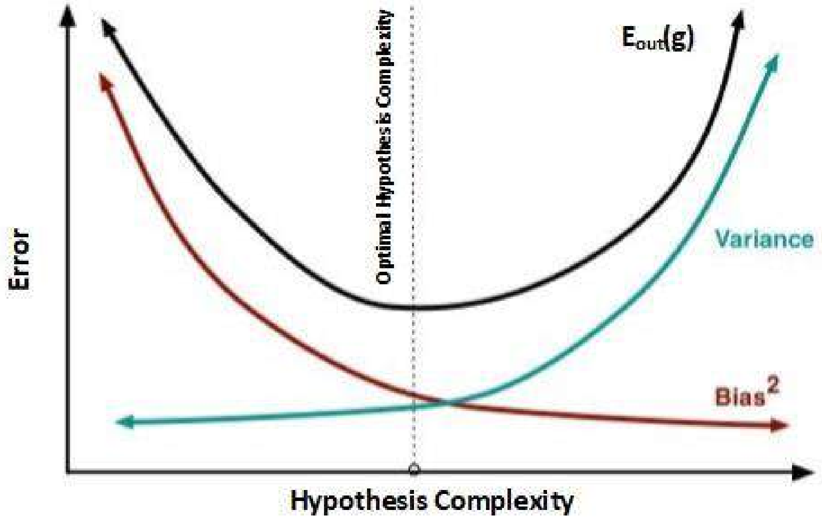


Figure 5.4: Contribution of the Bias and the Variance to the out of sample error as function of the hypothesis set complexity. Adjusted from *Scot Fortmann-Roe*, by S. Fortmann-Roe, 2012, Retrieved November 13, 2018, from http://scott.fortmann-roe.com/docs/BiasVariance.html.

[5] In subsection 5.3.5 the reason to split the training data into a training, validation and test set is explained.

In this subsection at first the concept of regularisation is explained. This tuning technique makes it possible to prevent overfitting by means of a regularisation parameter, $\lambda$. Subsequently, cross-validation is explained, which provides the tools required to duplicate the training data in order to optimise the model parameters and the regularisation parameter. Both techniques together enable the possibility of tuning the learning model.

### Regularisation

Regularisation is a tuning technique to correct overfitting. Instead of $E_{in}(g)$, it presents an augmented error, $E_{aug}(g)$, which is equal to $E_{in}(g)$ plus an "overfitting penalty". The overfitting penalty adds a penalty to the learning model parameters to reduce the freedom of fitting the stochastic noise.[6] The magnitude of the penalty depends on the chosen $\lambda$. In Figure 5.5 for different values of $\lambda$ the effect of regularisation is shown. In case $\lambda = 0$, there no effect of regularisation. It can be observed that a too large $\lambda$ results again in underfitting ($\lambda = 1$), however for the correctly chosen value ($\lambda = 0.0001$) overfitting can be fully corrected. Finding the correct $\lambda$ is an iterative process, which is performed through cross-validation.

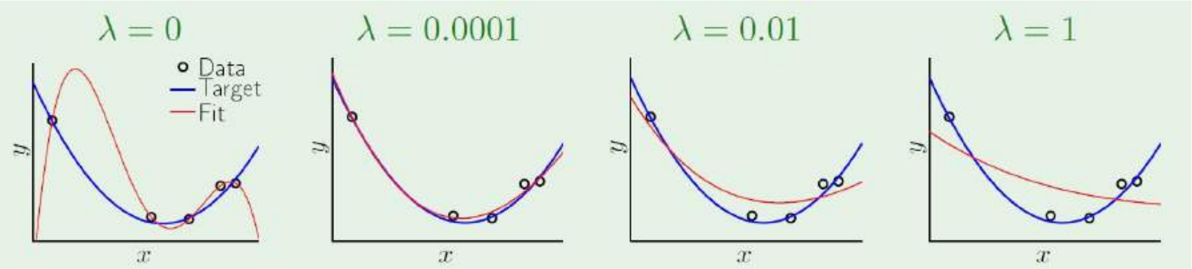


Figure 5.5: Visualisation of regularisation. Reprinted from *Learning From Data*, by Abu-Mostafa, 2012.

### Cross-Validation

Cross-validation enables duplication of the training data in order to perform the Bias-Variance Analysis and tuning the learning model, without violating the principle of generalisation. Before the principle of cross-validation is explained, at first the purpose of validation is explained by showing a correctly conducted learning process of a single hypothesis, where no human intelligence has been leaked into performance measure of the final hypothesis:

Assume the quantity of training data, which represents correctly the complexity of the input space, is $D$. Firstly, the training data is randomly split in a training set $N$ and a test set $T$. Secondly, the hypothesis is trained on the training set and tested on the test set. Lastly, the hypothesis is trained on all the training data together in order to derive the final hypothesis $f$. In terms of contamination, the training set is fully contaminated since it has been learned by the algorithm. Subsequently, the performance is tested on the undisturbed test set to obtain an unbiased error measure. The result is an unbiased $E_{out,\ test}(g)$ which guarantees the final hypothesis to generalise. Figure 5.6 visualises the steps of a correctly performed learning process of a single hypothesis.

[6] More detailed information about regularisation techniques and the corresponding mathematical derivations, can be found in *"Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance"* by Ng [33].

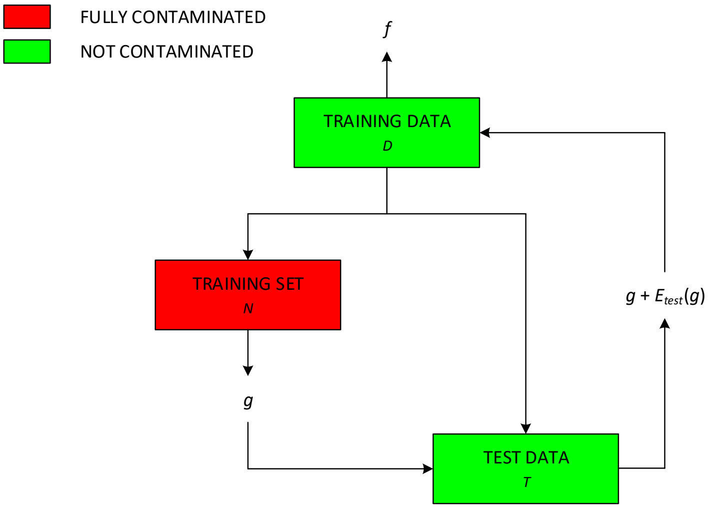


Figure 5.6: Visualisation of a correctly performed learning process of a single hypothesis. Adjusted from Learning From Data, by Abu-Mostafa, 2012.

In the explanation of regularisation it has already been mentioned that the search for the correct $\lambda$ value to minimise $E_{aug}(g)$, is an iterative process. This implies that a range of $\lambda$ values, $M$, has to be tested. This means that both $N$ and $T$ have to be divided into $M$ "folds" to train each hypothesis (a specific algorithm with a certain value of $\lambda$, where $\lambda \in M$), assuming the size of $D$ is capable of delivering $M$ training and test sets. The problem arises the moment the results of the trained hypotheses are compared to each other through testing on the test set folds. Per hypothesis the performance is measured by an error measure[7] and the value of $\lambda$ that returns the optimal Bias-Variance Trade-off, or in other words the smallest $E_{aug}(g)$, is selected as the best hypothesis according the Bias-Variance Analysis. By performing the Bias-Variance Analysis overfitting occurs by assuming the minimal $E_{aug}(g)$ reflects the real $E_{out}(g)$. After all, the choice for the best hypothesis is based on the performance of learning the training data, therefore it is not said that the pattern of the input space is learned. This means that testing $M$ hypotheses on $M$ test sets, results in pollution of the test set and consequently the error measurements do not longer report the generalisation performance.

In order to obtain a final hypothesis, $f$, that is guaranteed to generalise according to the $E_{out,\ test}(g)$, the Bias-Variance Analysis has to be performed prior to testing the best hypothesis on the test set. This implies the split of a third set: the validation set $K$. Instead of $T$, now $N$ and $K$ are split into $M$ folds. Again, each hypothesis is trained and tested on the folds and the best hypothesis is chosen based on results of the Bias-Variance Analysis. Subsequently, the best hypothesis is tested on the test set to deliver the unbiased $E_{out,\ test}(g)$. In this process the validation set is only used to decide which value of $\lambda$ return the best hypothesis, therefore it has become slightly contaminated by human intelligence, resulting in an optimistic bias. However, the test set remained untouched and is therefore capable to report the generalisation performance. Figure 5.7 visualises the steps that are included in a correctly performed learning process with multiple hypotheses.

[7] In subsection 7.4.1 more information is given regarding error measure methods.

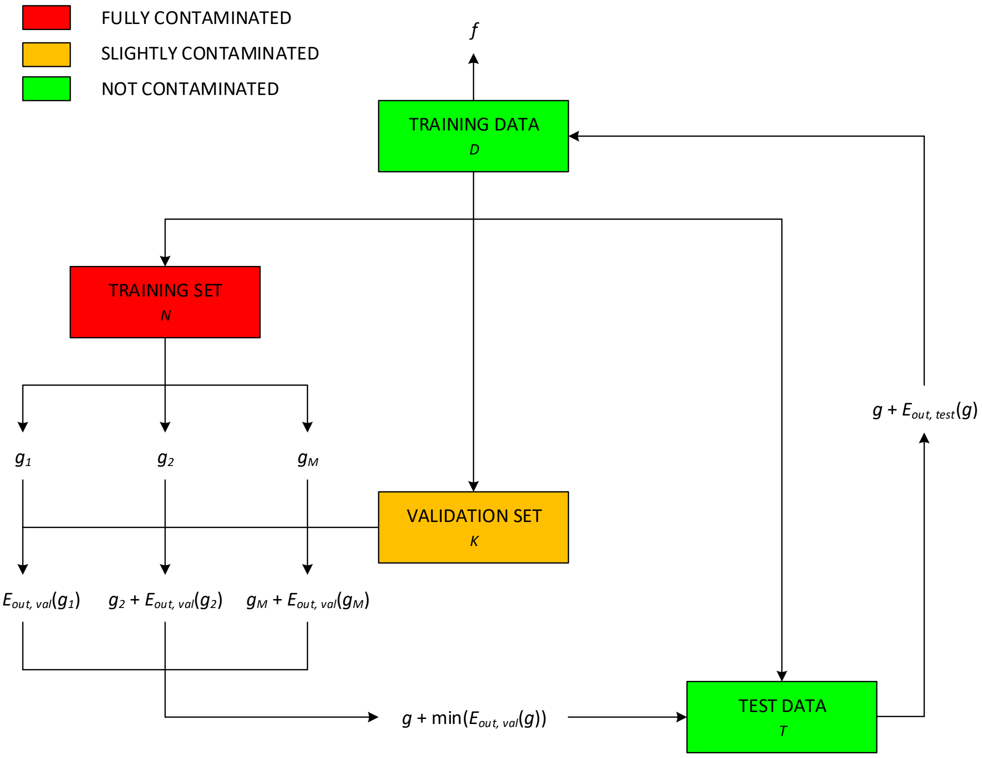


Figure 5.7: Visualisation of a correctly performed learning process with multiple hypotheses. Adjusted from *Learning From Data*, by Abu-Mostafa, 2012.

Although it generates an unbiased $E_{out,\ test}(g)$, the downside if this solution is that the training data is split in three parts ($T$, $N$ and $K$), of which subsequently $N$ and $K$ are split into folds to train and test each hypothesis. Figure 5.2b shows that a decreasing $N$ is unfavourable with respect to the Variance, resulting in an increasing $E_{out}(g)$. According to the Bias-Variance Trade-off, the split ratio itself does not affect $E_{out}(g)$.

The idea behind cross-validation is to provide both a large N and K, to enable the possibility of performing a proper Bias-Variance Analyses and to test if the found $E_{aug}(g)$ truly reflects the real $E_{out}(g)$. This can be acquired by using a cross-validation iterator, which uses the "leave one out" technique. This implies that after the test set has been randomly split from the training data, the remaining part $N$ is randomly split into $k$ folds (usually $k = 10$ is chosen). Subsequently, $N$ - $k$ data points are used for the training and $k$ data points for validation, which can be repeated $N$ / $k$ times. Figure 5.8 shows this process.

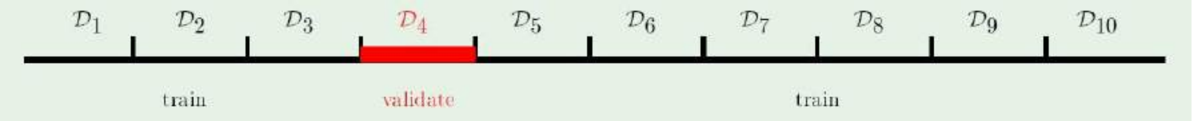


Figure 5.8: Visualisation of the leaving one out technique. Reprinted from Learning From Data, by Abu-Mostafa, 2012.

This process again can be extented by using random splitting methods. An example of such iterator is the StratifiedShuffleSplit, visualised in Figure 5.9. This method randomly splits the data $k$ times by a preferred ratio into $N$ and $K$, whereas the distribution of label classes or groups is preserved.

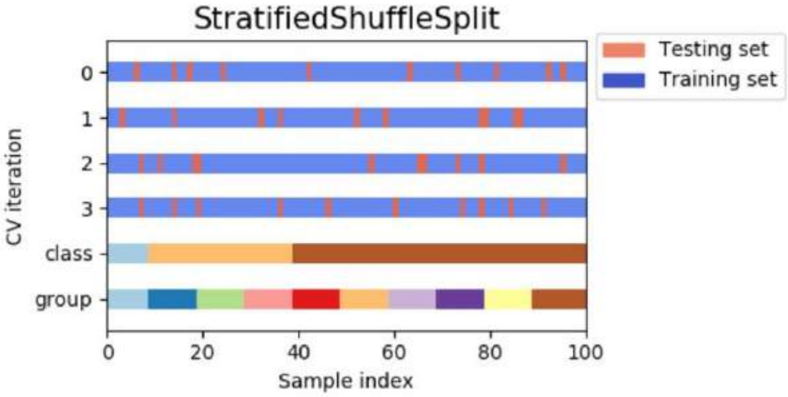


Figure 5.9: Visualisation of the cross-validation iterator StratifiedShuffleSplit. Reprinted from *scikit-learn*, n.d., Retrieved November 15, 2018, from `https://scikit-learn.org/stable/modules/cross_validation`.

## 5.4. Supervised Learning Algorithm

The list of algorithms capable of solving a Supervised Learning problem contains a lot of different algorithms, of which various adjustments are made, of which each variant itself can be seen as a new algorithm. Two different types of learning problems can be distinguished in Supervised Learning: classification and regression algorithms. As the name suggest, a classification algorithm predicts the output space into classes or class probabilities, while a regression algorithm predicts a continuous output. Some algorithms are specialised in one of the two learning problems, but some are capable of performing both tasks. In section 2.1 is mentioned that the learning problem concerns a classification of WBI-SOS subsoil scenarios. This reduces the number of possible algorithms, however still many promising algorithms remain left.

In Kotsiantis [28], the best-known classifications techniques for Supervised Learning, including explanation of the associated algorithms are discussed. An overview of the following classification techniques given:

- Logistic-based algorithms
- Perception-based techniques
- Statistical Learning algorithms
- Instance-based learning
- Support Vector Machines

According to Kotsiantis [28] there is no superior algorithm, but the conditions of the learning problem decide whether a specific algorithm is capable of outperforming the others. The conditions are depending on factors such as quantity and quality of training data or available training time, etc. The solution to find the best algorithm is to test multiple algorithms and by means of cross-validation detect the one that performs the best given a certain error measure criterium.

As mentioned in subsection 2.3.3, out of simplification only one classification technique is researched in this thesis. For this thesis it is decided to test the most recently developed classification technique for Supervised Learning: Support Vector Machines. Due to its high accuracy and the ability to deal with large number of features with respect the quantity of the training data [28][34], it offers opportunities to extend the feature extraction method[8], therefore making it easy to sync with further research on the learning problem as stated in this thesis. Furthermore, Support Vector Machines are capable of performing both classification and regression tasks, therefore in case one wants to change in a follow-up research the learning problem from a classification into a regression problem, the knowledge gained in this thesis can be used directly.

[8] More information regarding the feature extraction method is provided in chapter 6.

### 5.4.1. Support Vector Machines

Support Vector Machines (SVM) are developed by Vapnik and Lerner [47] and revolve around maximising a margin around the decision boundary between two separable data populations[34]. In this subsection the idea behind the SVM is conceptually explained.[9] First, the concept of a maximising a margin is explained by discussing the Linear SVM. Secondly, the Soft Margin is explained, which provides the opportunity to deal with overlapping of data populations.

#### Hard Margin Linear Support Vector Machines

Hard margin linear SVM can only be applied in case of two linear separable data populations. In the left plot of Figure 5.10 such linear separable situation is shown by the the red circles and the green plus signs. The question one can ask is which of the collection of possible decision boundaries, represented by the black dotted lines, separates the two population the best. The solution to this question is given in the right plot. As mentioned, a SVM chooses the best decision boundary based on the maximised margin that can be achieved given two linearly separable populations. It can be seen that the parallel "negative" and "positive" hyperplanes are fit between the two populations. The angle under which is it positioned at the moment gives the largest possible margin. Any other angle would decrease the margin. The data points that support both hyperplanes are accordingly called the support vectors. The final decision boundary is centered in between the negative and positive hyperplanes. During the learning process, the algorithm calculates for each possible decision boundary, the orthogonal distance, $w$, of all the associated support vectors that support the two parallel hyperplanes. The best decision boundary has the minimum $||w||$. Note, all other data points that are not assigned as support vectors have no influence on the best decision boundary.

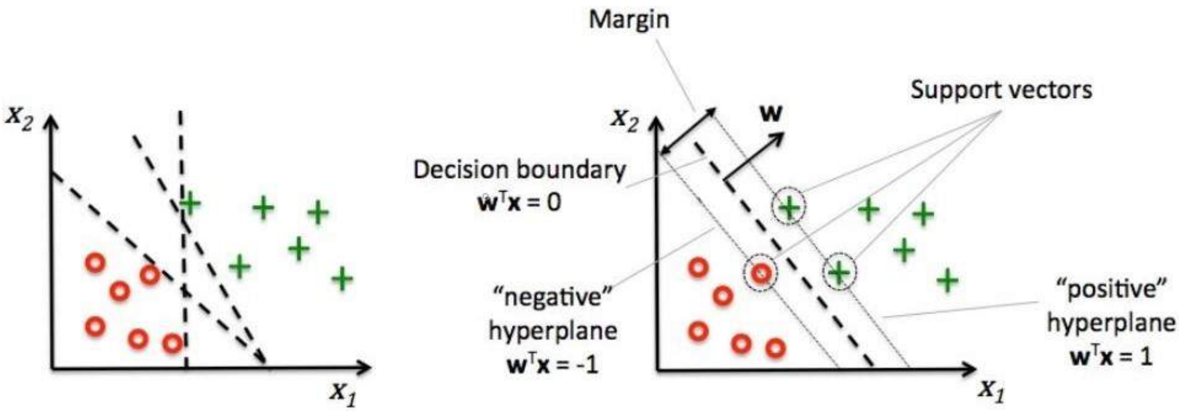


Figure 5.10: The concept of a linear Support Vector Machine. Adjusted from *Python Machine Learning,* by Raschka, 2015.

#### Soft Margin Linear Support Vector Machines

In case the the data populations are positioned as shown in the left plot of Figure 5.11, it is not possible to find a linear decision boundary that is capable of separating both populations. However, observing the positions of both populations, it is likely that the singe green plus is an outlier. In order to deal with data sets that are linearly separable, except for some outliers, the so called "soft margin" has been integrated in the SVM by Cortes and Vapnik [15], which is visualised in the right plot of 5.12. A soft margin accepts some outliers by giving them a penalty called the "slack", which is denoted by $\xi$. The magnitude of the slack depends on the orthogonal distance to the hyperplane parallel that is supported by the population associated the outlier. Now the best decision boundary has the minimum $1/2||w|| + \mathrm{C}\sum_{n=1}^{N}\xi_n$.

[9] For more information about the mathematical proof of the content is referred to *A Tutorial on Support Vector Machines for Pattern Recognitions* by Burges [13].

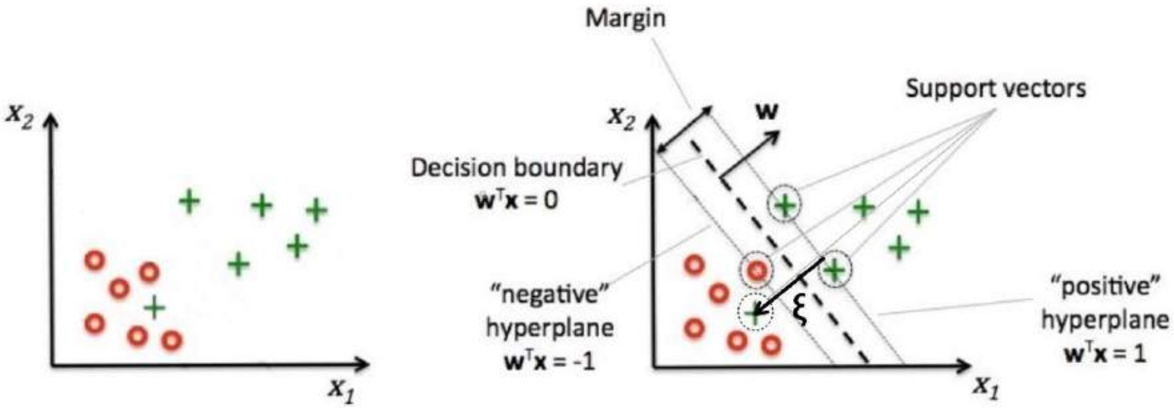


Figure 5.11: The concept of a soft margin linear Support Vector Machine.[10]Adjusted from *Python Machine Learning*, by Raschka, 2015.

### Non-Linear Support Vector Machines

Although the integration of a Soft Margin increased the application range for the SVM, the development of the Non-Linear SVM by Boser et al. [11] made it possible to deal directly with non-linearly separated data populations without preprocessing the data set. The authors implemented the so called "kernel trick". Figure 5.12 shows the principle of the kernel trick. Through transforming a non-linear separable feature space into a higher dimensional space, one can obtain a linear separable feature space. Often used kernels to perform the transformation are the Polynomial kernel, the Gaussian Radial Basis Function kernel and the Sigmoid kernel.

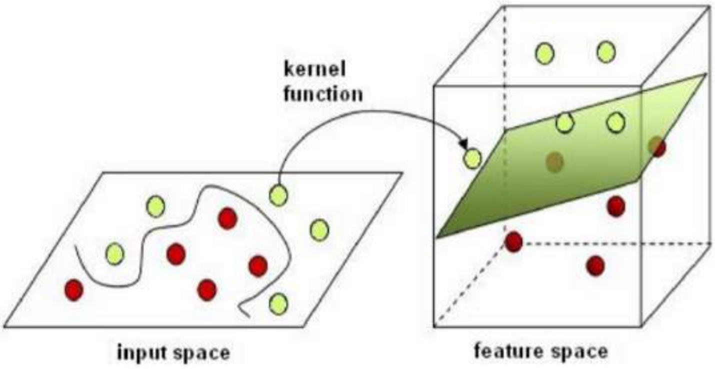


Figure 5.12: The principle of the kernel trick. Reprinted from *Bagging Support Vector Machines for Leukemia Classification*, by Zararsiz et al., 2012.

Like the statement made by Kotsiantis [28] regarding the choice for a certain algorithm, also both Theodoridis and Koutroumbas [45] and Ben-Hur and Weston [9] state there is no practical method for the best selection of the kernel function. Again, the solution to find the best kernel is to test multiple kernels and by means of cross-validation detect the one that performs the best given a certain error measure criterium. To comply with the assumption of testing only one classification technique as mentioned in subsection 2.3.3, only the Gaussian Radial Basis Function kernel (Gaussian RBF kernel) is applied in this thesis. This kernel is mostly applied in the absence of prior knowledge of the data distribution[5], and therefore this choice relates with the argumentation in section 5.2 of choosing a Passive Teacher for the learning problem. This decision for the Gaussian RBF kernel is strengthened by the experience of Ben-Hur and Weston [9] and the findings of Yekkehkhany et al. [52], where multiple kernels are compared to classify multi-temporal polarimetry SAR data, which showed that the accuracies of SVM equipped with a Gaussian RBF kernel for various crop types are relatively better than the Linear and the third degree Polynomial kernel functions.

[10] The outlier in this figure is incorrectly marked as a support vector. The reason behind this deliberate deception is to comply the presented theory with the output of the Python package *scikit-learn* [2], as shown in Figure 7.9.

# III

# APPLICATION & IMPLEMENTATION

# 6

# THE FEATURE EXTRACTION METHOD

In this chapter the feature extraction method is developed, through which features of the compressed soil column are obtained, given a settlement curve and the corresponding load information. Firstly, in section 6.1 the required information to develop the feature extraction method is summarised. Secondly, in section 6.2 the development process is explained and lastly in section 6.3 accuracy of the developed feature extraction method is investigated.

## 6.1. Required Information

Being able to distinguish WBI-SOS subsoil scenario, the feature extraction method has to deliver unique features per subsoil scenarios. As is it commonly known that the magnitude of the load is directly related to the magnitude of the settlement, in order to represent only the compressed soil column, the feature extraction method is deemed to standardise any feature with respect to any load combination that can be expected within the simulated heightening case. The data used to develop the feature extraction method is the distributed load information and the subsoil scenarios as presented by the median values of the layer interface distributions. During the development of the subsoil scenarios, the available data has been divided into distinguishable groups. Each subsoil scenario is the statistical representation of each group, of which the group mean is presented by the medians of the layer interface distributions. Hence, they form the best possible reference points to develop a feature extraction method that is capable of distinguishing individual subsoil scenarios. An overview of this data is shown in Figure 6.1.

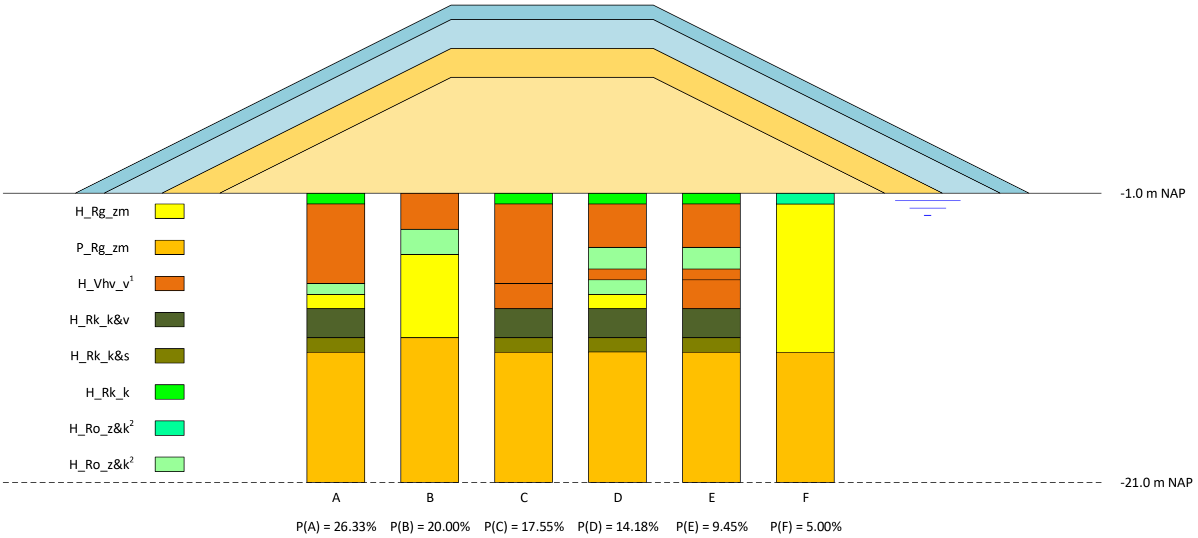


Figure 6.1: Visualisation of the required information from the simulated heightening case.

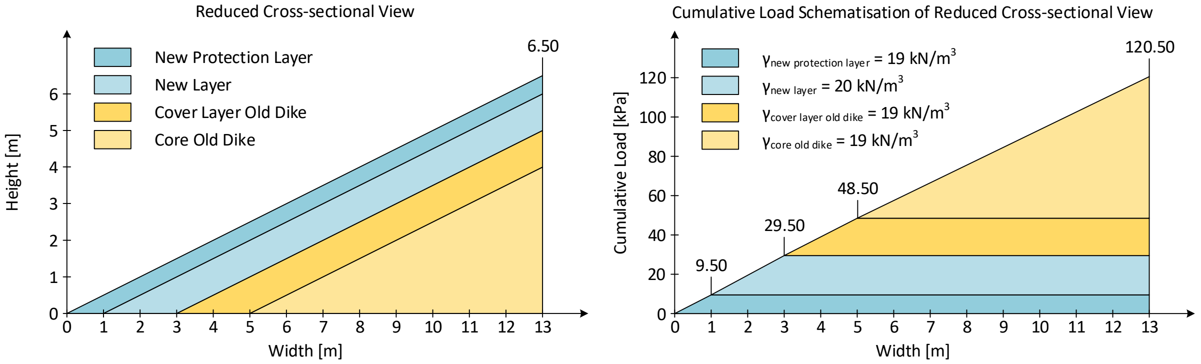
## 6.2. Development of the Feature Extraction Method

In this section the development of the feature extraction method is explained. With help of this method combinations of settlement and loading curves are characterised into features that represent the compressed soil column.

### 6.2.1. Influence of the Load

Settlement of the subsoil is caused by compression of a soil column due to a new load applied on top of the column. Purely seen from the load perspective, the higher the magnitude of the new load, the more settlement occurs. Besides the magnitude of the new load, the magnitude of the preload of the subsoil plays an important role. The more the soil has been preloaded in the past, the less is will consolidate compared to a non-preloaded soil column subjected to the same new load. To find the features that represent only the compressed soil column, the influence of both phenomena on the measured settlement curve has to be understood.

The first task is to define the expected load combinations of new load and preload by the old dike body. Given the cross-sectional overview presented in Figure 6.1, due to symmetry, one can observe that all possible load combinations can be extracted from the reduced cross-sectional overview presented in Figure 6.2a. From this overview a cumulative load schematisation is constructed, shown in Figure 6.2b. In this figure, starting at 0.25 m, with an interval of 0.50 m and at 1.00 m, 3.00 m, 5.00 m and 13.00 m the load combinations are calculated.[1] The result are 30 different load combinations.



(a) Reduced cross-sectional overview which summarises all possible load combinations.

(b) Cumulative load schematisation of the reduced cross-sectional overview used to define load combinations.

Figure 6.2: Derivation of the cumulative load schematisation from the reduced cross-sectional overview.

In order to get an idea about the influence of the different 30 load combinations, settlement curves over a period of 100 000 days[2] have been calculated per WBI-SOS subsoil scenario by the integrated a,b,c-isotache model of D-Settlement. The subsoil scenario input are the subsoil scenarios presented by the median values of the layer interface distributions. In Figure 6.3 the outcome of these calculations are visualised. One can see that subsoil scenario F can directly be distinguished. The overlap between the others subsoil scenarios is tremendous and shows that the influence of the load has to be eliminated.

[1] The interval of 0.50 m is chosen to represent surface settlement measurement from a grid of 0.25 m$^2$ [42]. The load combinations derived at 1.00 m, 3.00 m, 5.00 m and 13.00 m are included because at these locations each layer has reached its maximum thickness.

[2] This extremely long period is chosen to be sure that every obtained settlement curve has reached its secondary consolidation stage.

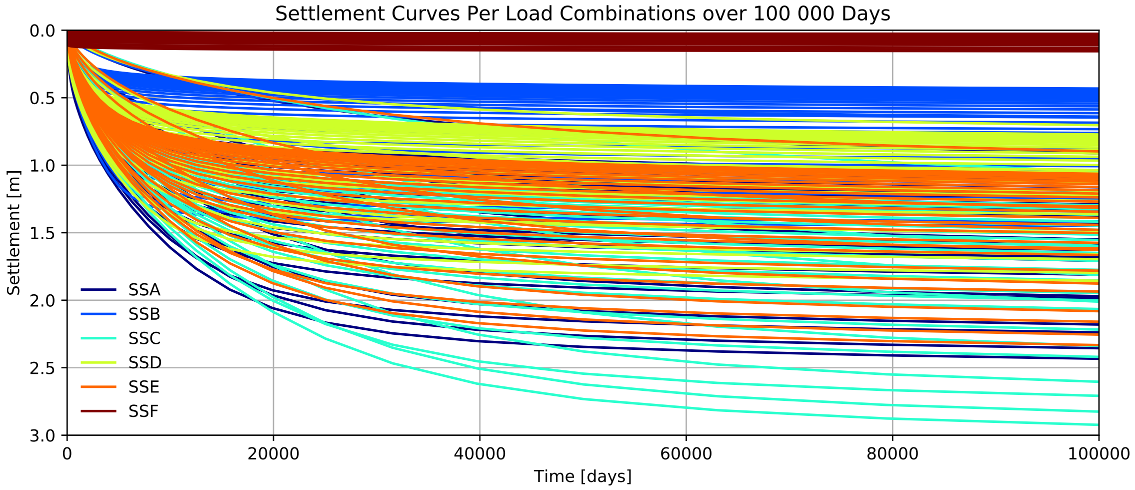


Figure 6.3: Influence of variation in load combinations per WBI-SOS subsoil scenario (without layer interface distributions).

Figure 6.4 shows the final settlement (after 100 000 days) of the settlement curves which are shown in Figure 6.3, categorised per load combination.

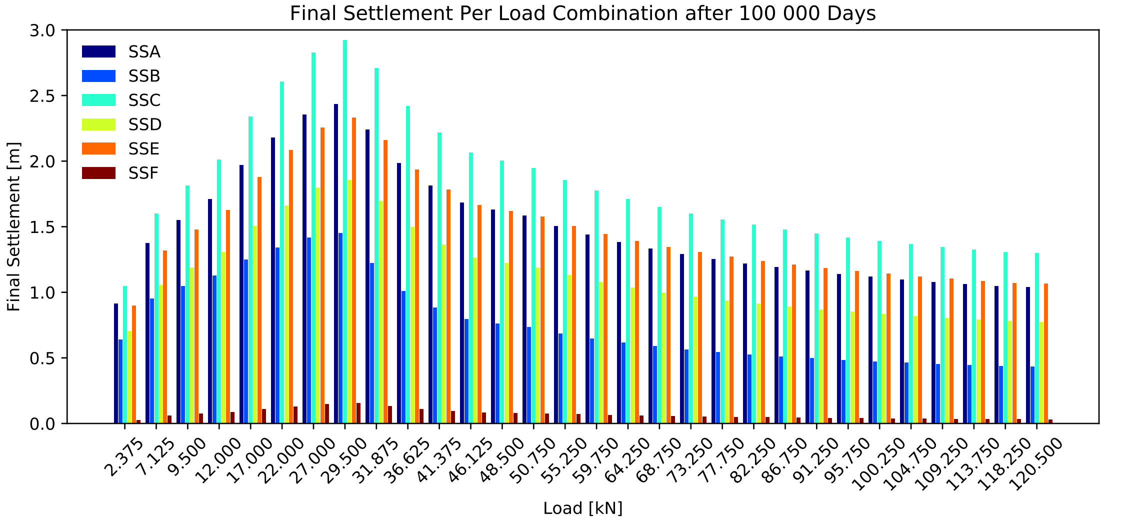


Figure 6.4: Settlement variation over the WBI-SOS subsoil scenarios (without layer interface distributions) per load combination.

From Figure 6.4 it can be concluded that each subsoil scenarios is unique for any load combination. This proves that it is possible to distinguish subsoil scenarios. Also, it can be seen that for any load combination the settlement of subsoil scenario F is distinctive from the others. Since its associated soil column is almost completely composed of sand, the settlement remains minimal. Therefore, it can be concluded that the spatial distribution of subsoil scenario F can directly be located from spatial surface settlement measurements, without the need of a Machine Learning application. Furthermore, by analysing the composition of subsoil scenario B, it can be seen that its associated soil column is mostly composed out of sand. The difference in magnitude of the final settlement with the other scenarios is also significantly. Given the spread of the layer interface distribution of its sand layer, it is expected that this difference remains sufficient. Also the gradient of its secondary consolidation stage (see Figure 6.6) deviates from the other scenarios due to the relative small quantity of compressible material.[3] Only for the load combinations that contain solely a new load component (2.375 kN to 29.500 kN), it shows similar behaviour with subsoil scenario D. By observing the cumulative load schematisation of the reduced cross-sectional area, it can be seen that these load combinations occur in approximately the first three meters. This implies that either the application range of the Machine Learning

[3] The feature extraction method developed in this thesis only extracts two features from the primary consolidation stage, therefore this argument is not relevant. However, in case one plans to increase the number of features by including feature extraction from the secondary consolidation stage, this argument becomes valid.

application that is developed in thesis reduces to the part of the cross-section that delivers unique settlements for subsoil scenarios B or that it is accepted that in the part of the cross-section where the uniqueness between the subsoil scenarios B and D is not guaranteed, the reliability of the classification between the subsoil scenario B and D strongly reduces.[4] Based on these arguments, in this thesis it is chosen to assume that also the spatial distribution of subsoil scenario B is directly distinguishable from spatial surface settlement measurements, without the need of a Machine Learning application.

### 6.2.2. Fitting the Settlement Curves

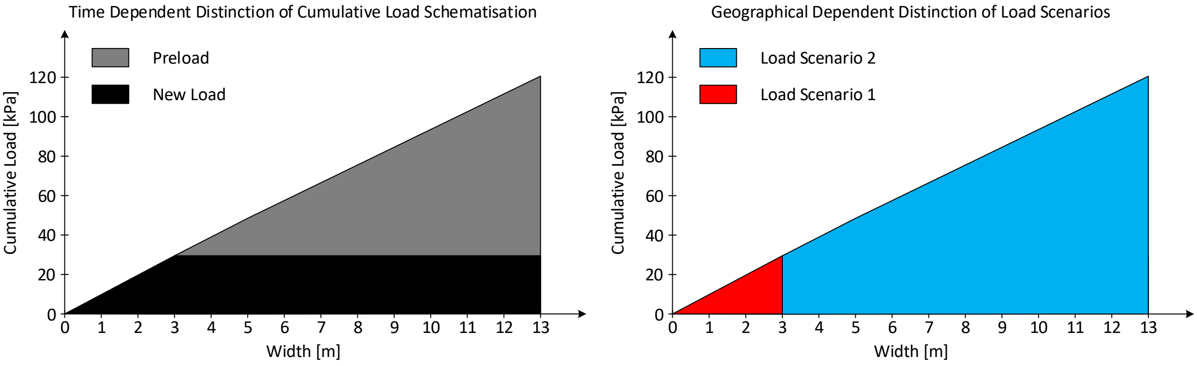
Before any characteristics can be extracted from the settlement curves, at first a method is required to fit the primary consolidation stage of the settlement curves.[5] Jiang et al. [23] describes six fitting models that have shown promising results for predicting behaviour of complex soil formations. The Logarithmic Model, the Power Model, the Hyperbolic Model, the Logistic Model, the Compertz Model and the Combined Gompertz-Logistic Model. After testing each model, it turned that only the Power Model was capable of fitting the predicted settlement curves, calculated by the integrated a,b,c-isotache model of D-Settlement. In Equation 6.1 the Power Model equation is shown.

$$s = Fit\,Constant\;a \cdot t^{\text{Fit Constant b}} \tag{6.1}$$

where:

| | | | |
|---|---|---|---|
| $s$ | = | Settlement | [m] |
| $t$ | = | Time | [days] |
| $Fit\,Constant\;a$ | = | *Fit Constant a* | [-] |
| $Fit\,Constant\;b$ | = | *Fit Constant b* | [-] |

Figure 6.6 shows two possible ways to distinguish the cumulative load schematisation of the reduced cross-section overview, illustrated in Figure 6.2b. Figure 6.5a presents a distinction between the new load and the preload and Figure 6.5b shows the composition of the load scenarios. Load scenario 1 describes an increasing new load and load scenario 2 describes a constant maximum load plus an increasing preload. In this thesis it was preferred to work with load scenarios, which underlying reasons will be clarified in the discussion of the figures 6.7 and 6.8.



(a) Distinction between the new load and the preload.

(b) Distinction between load scenario 1 and 2.

Figure 6.5: Definition of the load scenarios from the cumulative load schematisation of the reduced cross-sectional overview.

In Figure 6.6 the results of the fit is shown per WBI-SOS subsoil scenario (without layer interface distribution), each subjected to the 30 load combinations. Expressed in red and blue are the calculated settlement curves per load scenario. In each of the sub-figures the predicted settlement curves have been plotted on a logarithmic time scale. This reveals the gradient change which indicates the transition from the primary into the secondary consolidation stage. This point is highlighted with a coloured star at the intersection of the vertical green bar with the corresponding settlement curve. As mentioned before, to simplify the research, in this thesis only the primary consolidation part is being fit, shown by the coloured dots.

[4] Given the decision the create classes of subsoil scenarios as explained in subsection 7.3.3, the latter would not result in a problem. Class 1 is formed by subsoil scenario A and D, which is characterised for containing a sand layer (for example to target the primary failure mechanism piping). As subsoil scenario B is mostly contained of sand, an incorrect classification between B and D is not harmful.

[5] The reason to fit only the primary consolidation stage is solely to simplify the research. The underlying reasons are explained in section 10.2.

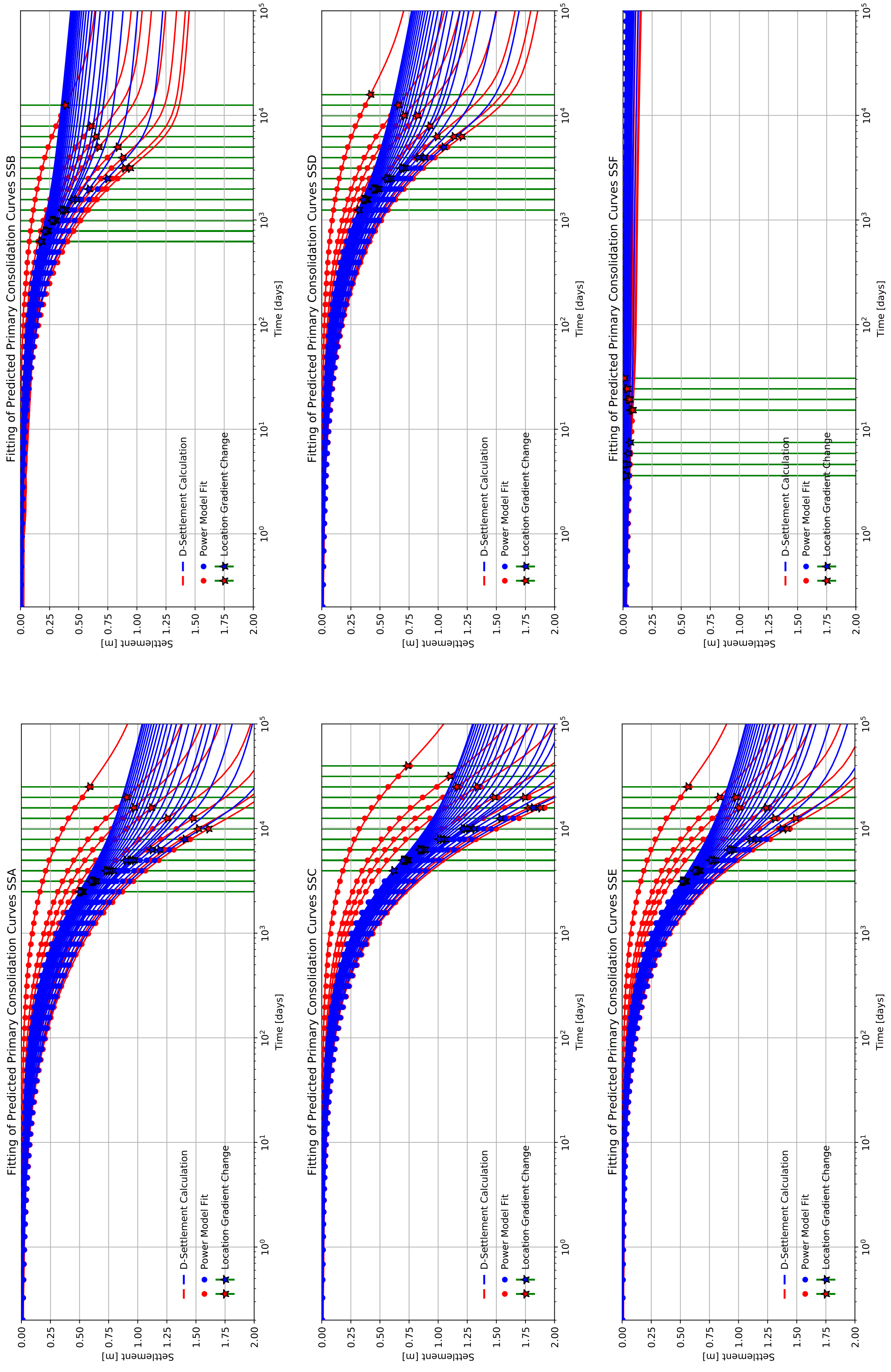


Figure 6.6: Fitting results of the Power Model on the settlement curves predictions per WBI-SOS subsoil scenario (without layer interface distributions), subjected to the 30 load combinations.

In the figures 6.7 and 6.8 the fit results of the subsoil scenarios A, C, D and E are shown. Table A.1 of appendix A presents the associated data per fit. Load scenario 1 and load scenario 2 are presented in colours ranging respectively between red and yellow and blue and green.

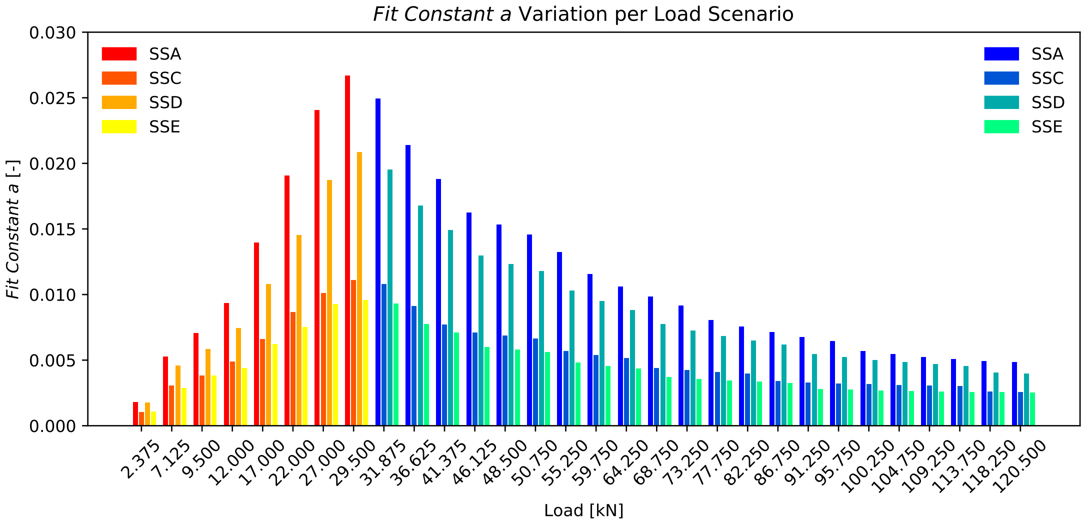


Figure 6.7: Variation in *Fit Constant a* per load scenario for each WBI-SOS subsoil scenario (without layer interface distributions).

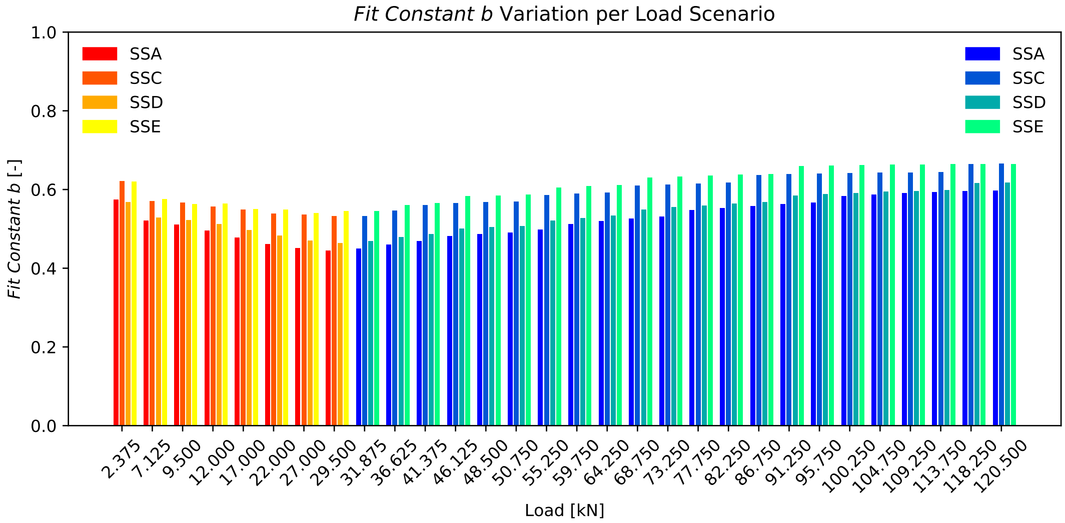


Figure 6.8: Variation in *Fit Constant b* per load scenario for each WBI-SOS subsoil scenario (without layer interface distributions).

The reason of using the defined load scenarios becomes clear through analysing the figures 6.7 and 6.8. The pattern that can be observed per load scenario for *Fit Constant a*, matches the relations which were already mentioned in the introduction of this section:

- The higher the new load, the more settlement occurs.
- The more the soil has been preloaded, the less is will consolidate.

Therefore, from a physical perspective, *Fit Constant a* could be explained as the depth average load dependent coefficient of compressibility of the primary consolidation stage. With *Fit Constant a* being the depth averaged load dependent coefficient of compressibility, *Fit Constant b* can be physically explained as the depth averaged load dependent coefficient of vertical permeability of the primary consolidation. Note, although a physical description exists for both fit constants, units are not assigned. The nature of classification by a Machine Learning algorithm does not value units (the algorithm learns purely by itself based on the provided training data). To stimulate this concept, by purpose no units are assigned.

### 6.2.3. Extracting the Features

Given the observed patterns for both fit constants per load scenario in the figures 6.7 and 6.8, it should be possible to extract features that express the compressed soil columns which are underlying to the settlement predictions. To find these features, per fit constant two formulas are created (one per load scenario), which should reproduce the pattern per load scenario. All the formulas are subjected to the following constrains:

- Because the predicted settlement curves per subsoil scenario originate from the same soil column, the variable that expresses the feature, has to be constant for each subsoil scenario for both load scenarios.
- Since all patterns are related to the load scenarios, only the magnitude of the new load and the preload are variable.

Given these constrains, a formula was simultaneously fit to all the derived fit constant values per load scenario. The open source Python library *LmFit* [1] is used to conduct the fits. Except for the magnitude of the new and the preload, all other parameters were fully constrained, which means that they could only take one value. Since all the load combinations associated to load scenario 2 include the maximum vertical stress of the new load value ($\sigma_{new\ load,\ max}$), the patterns for both fit constants meet at the transit of the load scenarios (29.500 kN - 31.875 kN). Therefore, at first the formulas for both fit constants were fit to load scenario 1, in order to find $x_{\text{feature 1}}$ and $x_{\text{feature 2}}$ per scenario and the corresponding constrained constants. Subsequently, the outcome of these fits were used as input to fit load scenario 2. Hence, the first term for the both equations 6.3 and 6.5 includes the constrained constants which were fit in respectively Equation 6.2 and Equation 6.4 and $\sigma_{new\ load,\ max}$. The results of the fit to load scenario 1 for *Fit Constant a* is presented by Equation 6.2.

$$Fit\ Constant\ a_{load\ scenario\ 1,\ i} \quad = \quad x_{feature\ 1} \cdot \sigma_{new\ load,\ i}{}^{c_1} \tag{6.2}$$

where:

| | | | |
|---|---|---|---|
| $x_{feature\ 1}$ | = | Fitted feature 1 | [-] |
| $Fit\ Constant\ a_{load\ scenario\ 1,\ i}$ | = | *Fit Constant a* of load combination i, part of load scenario 1 | [-] |
| $\sigma_{new\ load,\ i}$ | = | Vertical stress of the new load of load combination i | [kPa] |
| $c_1$ | = | Constrained constant 1 (Equation 6.2) | [-] |

The results of the fit to load scenario 2 for *Fit Constant a* is presented by Equation 6.3, in which the outcome of Equation 6.2, ($x_{\text{feature 1}}$ and $c_1$) is used as input.

$$Fit\ Constant\ a_{load\ scenario\ 2,\ i} \quad = \quad x_{feature\ 1} \cdot \sigma_{new\ load,\ max}{}^{c_1} - x_{feature\ 1}{}^{c_2} \cdot c_3 \cdot ln\left(\frac{\sigma_{preload,\ i}}{c_4}\right) \tag{6.3}$$

where:

| | | | |
|---|---|---|---|
| $x_{feature\ 1}$ | = | Fitted feature 1 (Equation 6.2) | [-] |
| $Fit\ Constant\ a_{load\ scenario\ 2,\ i}$ | = | *Fit Constant a* of load combination i, part of load scenario 2 | [-] |
| $\sigma_{new\ load,\ max}$ | = | Maximal vertical stress of the new load | [kPa] |
| $\sigma_{preload,\ i}$ | = | Vertical stress of the preload of load combination i | [kPa] |
| $c_1$ | = | Constrained constant 1 (Equation 6.2) | [-] |
| $c_2$ | = | Constrained constant 2 (Equation 6.3) | [-] |
| $c_3$ | = | Constrained constant 3 (Equation 6.3) | [-] |

The results of the fit to load scenario 1 for *Fit Constant b* is presented by Equation 6.4.

$$Fit\ Constant\ b_{load\ scenario\ 1,\ i} \quad = \quad x_{feature\ 2}{}^{c_1} - x_{feature\ 2} \cdot ln\left(\sigma_{new\ load,\ i} + x_{feature\ 2} \cdot c_2\right) \tag{6.4}$$

where:

| | | | |
|---|---|---|---|
| $x_{feature\ 2}$ | = | Fitted feature 2 | [-] |
| $Fit\ Constant\ b_{load\ scenario\ 1,\ i}$ | = | *Fit Constant b* of load combination i, part of load scenario 1 | [-] |
| $\sigma_{new\ load,\ i}$ | = | Vertical stress of the new load of load combination i | [kPa] |
| $c_1$ | = | Constrained constant 1 (Equation 6.4) | [-] |
| $c_2$ | = | Constrained constant 2 (Equation 6.4) | [-] |

The results of the fit to load scenario 2 for *Fit Constant b* is presented by Equation 6.5, in which the outcome of Equation 6.4 ($x_{\text{feature 1}}$, $c_1$ and $c_2$) is used as input.

$$\begin{aligned} Fit\ Constant\ b_{load\ scenario\ 2,\ i} &= x_{feature\ 2}{}^{c_1} - x_{feature\ 2} \cdot ln\left(\sigma_{new\ load,\ max} + x_{feature\ 2} \cdot c_2\right) \\ &+ x_{feature\ 2} \cdot c_3 \cdot ln\left(\sigma_{preload,\ i} \cdot x^{x_{\text{feature 2}} \cdot c_4} + x_{feature\ 2} \cdot c_5\right) \end{aligned} \tag{6.5}$$

where:

| | | | |
|---|---|---|---|
| $x_{feature\ 2}$ | = | Fitted feature 2 (equation 6.4) | [-] |
| $Fit\ Constant\ b_{load\ scenario\ 2}$ | = | *Fit Constant b* of load combination i, part of load scenario 2 | [-] |
| $\sigma_{new\ load,\ max}$ | = | Maximal vertical stress of the new load | [kPa] |
| $\sigma_{preload,\ i}$ | = | Vertical stress of the preload of load combination i | [kPa] |
| $c_1$ | = | Constrained constant 1 (Equation 6.4) | [-] |
| $c_2$ | = | Constrained constant 2 (Equation 6.4) | [-] |
| $c_3$ | = | Constrained constant 3 (Equation 6.5) | [-] |
| $c_4$ | = | Constrained constant 4 (Equation 6.5) | [-] |
| $c_5$ | = | Constrained constant 5 (Equation 6.5) | [-] |

Table 6.1 shows the fit results of the formulas, including the associated coefficients of variation ($CV$).[6] Note, since some of the constrained constants found in Equation 6.2 and Equation 6.4 are used as input for Equation 6.3 and Equation 6.5, not all constrained constants are accompanied by a fit error.[7]

Table 6.1: Fit results of the formulas expressed by the equations 6.2, 6.3, 6.4 and 6.5

| | **Equation 6.2** | | **Equation 6.3** | | **Equation 6.4** | | **Equation 6.5** | |
|---|---|---|---|---|---|---|---|---|
| | **Fit value** | ***CV*** | **Fit value** | ***CV*** | **Fit value** | ***CV*** | **Fit value** | ***CV*** |
| **Constant** | [-] | [%] | [-] | [%] | [-] | [%] | [-] | [%] |
| $c_1$ | 1.09 | 2.03 | 1.09 | - | $1.23 \cdot 10^{-1}$ | 1.78 | $1.23 \cdot 10^{-1}$ | - |
| $c_2$ | - | - | 1.05 | 1.62 | 68.91 | 11.94 | 68.91 | - |
| $c_3$ | - | - | 13.23 | 12.93 | - | - | 1.36 | 3.09 |
| $c_4$ | - | - | 2.73 | 0.43 | - | - | 16.88 | 2.13 |
| $c_5$ | - | - | - | - | - | - | 11.77 | 4.06 |

In the figures 6.9 and 6.10 the fit accuracy of the four formulas over both load scenarios is visualised.

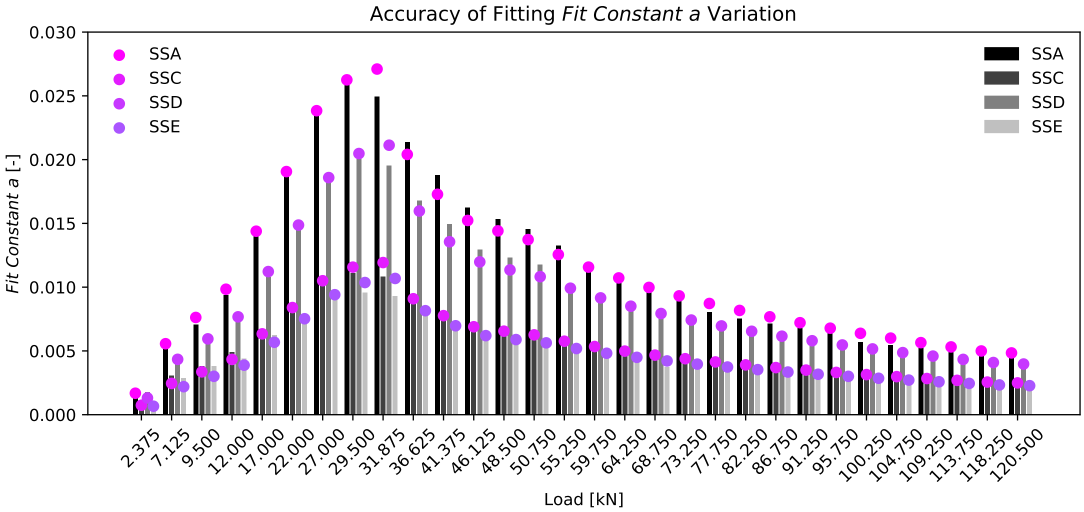


Figure 6.9: Visualisation of the fit accuracy of the formulas expressed by the equations 6.2 and 6.3.

[6] Because the quality of the feature extraction method is discussed in section 6.3, the fit results of $x_{feature\ 1}$ and $x_{feature\ 2}$ are shown in Table 6.2.

[7] The effects of excluding these fit errors is discussed in subsection 9.2.3.

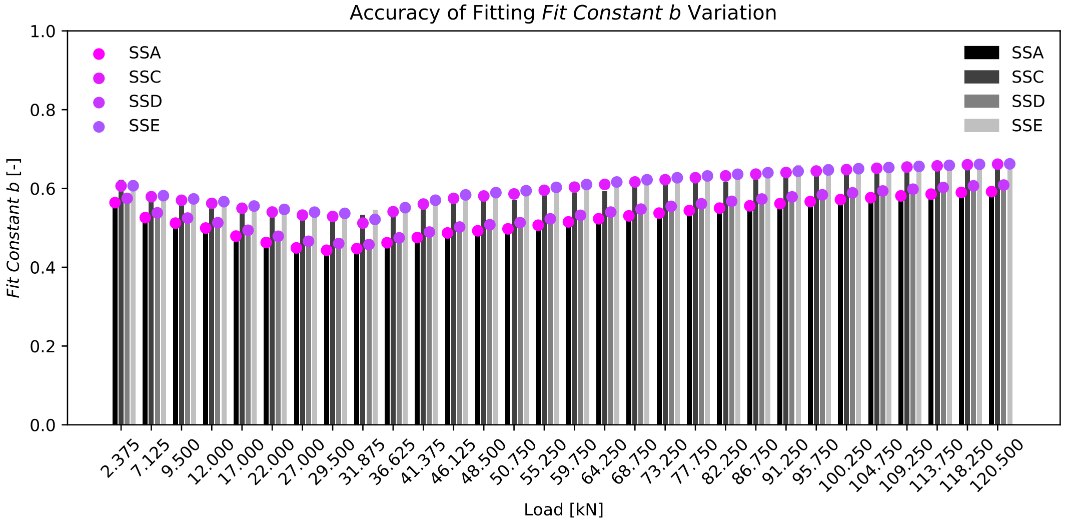


Figure 6.10: Visualisation of the fit accuracy of the formulas expressed by the equations 6.4 and 6.5.

## 6.3. Accuracy of the Feature Extraction Method

Since the feature extraction method consist of four empirical formulas, it is important to estimate the accuracy of the formulas. The first indication is acquired by the relative standard error (*RSE*) of feature 1 and 2 ($x_{feature\ 1}$ and $x_{feature\ 2}$), which were found during the fit of the four formulas. A second indication is found by using the developed feature extraction method to calculate the features for both the *Fit Constant a* and *Fit Constant b* per subsoil scenario and subsequently determine the coefficient of variation (*CV*) with respect to $x_{feature\ 1}$ and $x_{feature\ 2}$ that were found during the development process of the feature extraction method (see subsection 6.2.3). The variation of the calculated features with respect to $x_{feature\ 1}$ and $x_{feature\ 2}$ is visualised respectively the left and right plot of Figure 6.11.

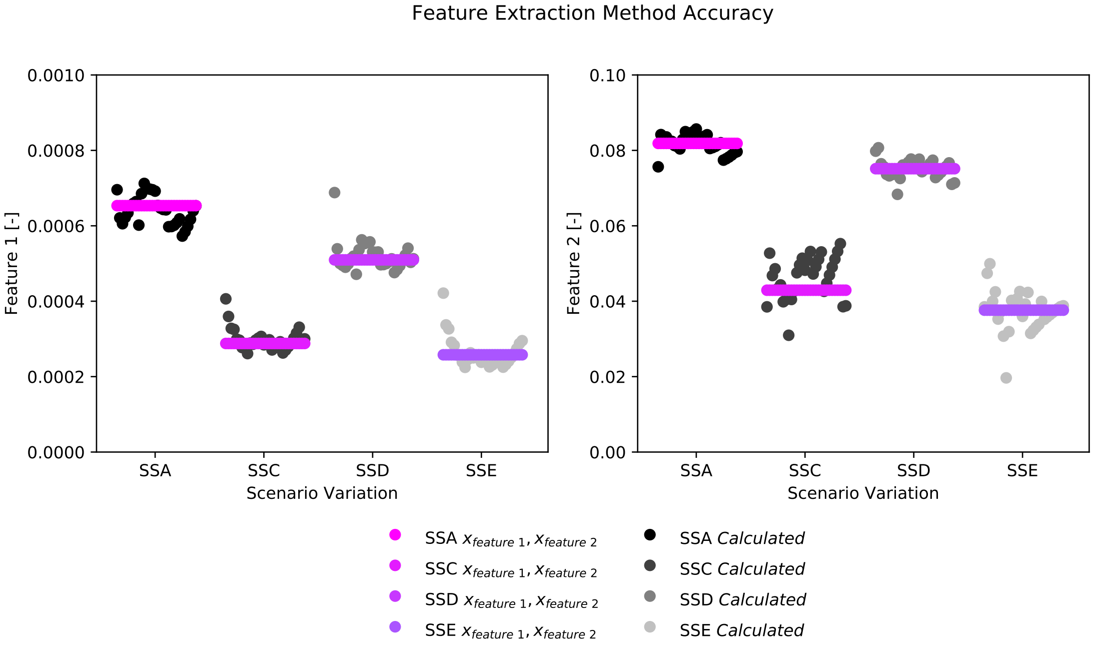


Figure 6.11: Visualisation of the variation of the calculated features, calculated by the feature extraction method, with respect to $x_{feature\ 1}$ and $x_{feature\ 2}$.

The results of the accuracy estimations are shown in Table 6.2. Table A.2 of appendix A presents the underlying data used to determine the coefficient of variation (*CV*) per feature.

Table 6.2: Overview of the accuracy estimations used to determine the accuracy of the developed feature extraction method.

| | **Feature 1** | | | **Feature 2** | | |
|---|---|---|---|---|---|---|
| | $x_{feature\ 1}$ | ***RSE*** | ***CV*** | $x_{feature\ 2}$ | ***RSE*** | ***CV*** |
| **WBI-SOS subsoil scenario** | [-] | [%] | [%] | [-] | [%] | [%] |
| A | $6.53 \cdot 10^{-4}$ | 7.08 | 5.87 | $8.19 \cdot 10^{-2}$ | 2.15 | 3.05 |
| C | $2.88 \cdot 10^{-4}$ | 7.38 | 10.08 | $4.29 \cdot 10^{-2}$ | 9.29 | 13.17 |
| D | $5.10 \cdot 10^{-4}$ | 7.13 | 7.70 | $7.51 \cdot 10^{-2}$ | 2.50 | 3.34 |
| E | $2.58 \cdot 10^{-4}$ | 7.47 | 15.55 | $3.76 \cdot 10^{-2}$ | 13.37 | 14.32 |

From Table 6.2 is can be seen that the RSE of both the fitted $x_{feature\ 1}$ and $x_{feature\ 2}$ are of the same order as the $c_v$ of the calculated features. This proves that both indications are reliable in providing an accuracy estimation of the feature extraction method, which is approximately 92%[8]. Typically for classification tasks, the performance of the feature extraction method is verified in combination with the quality of the learning process in terms of the prediction accuracy of the algorithm, since factors like the quantity of training data, the type of algorithm play and tuning techniques also have a strong influence on the prediction accuracy. Therefore, it is hard to state if the accuracy of the feature extraction method is sufficient or not.

In Resalat and Saba [37], the performance of multiple feature extraction methods have been tested in extracting feature sets from movement imagination data sets. Feature extraction methods accuracies of over 75% have been reached. In comparison to these numbers, the accuracy of the developed feature extraction method developed in this chapter is deemed to be sufficient.

Finally, in Figure 6.12 the relation between the calculated features is plotted. Since only two features are extracted by the feature extraction method, this plot is directly the feature space that will be analysed by the Machine Learning algorithm in order to create the preferred output space. From the overlap that is observed in the feature space, it is most likely that, given the accuracy of the feature extraction method in combination with only two extracted features, the preferred target complexity of the input is too high to provide an output space in terms of WBI-SOS subsoil scenarios. However, two distinguishable classes of subsoil scenarios can be clearly distinguished, of which class 1 and 2 are formed respectively by the subsoil scenarios A, D and C, E. In section 7.3.3 classification of the feature space is further elaborated.

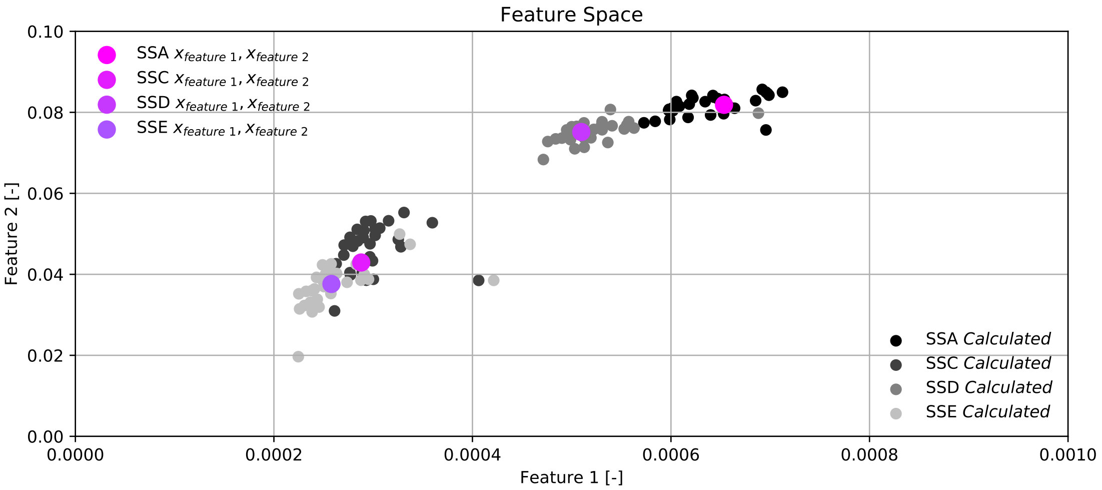


Figure 6.12: Visualisation of the feature space.

[8] The following formula is used to estimate the accuracy of the feature extraction method: 1 - 0.5 ($\overline{CV}_{feature\ 1} + \overline{CV}_{feature\ 2}$).

# 7

# THE LEARNING PROCESS

In section 5.4 it is explained that a Support Vector Machine (SVM) will be used for the Machine Learning application. In this chapter the learning process of training a SVM and validating the training results is presented. Firstly, in section 7.1 the required information to perform training is summarised. Secondly, in section 7.2 the environment is created through transforming the calculated synthetic surface settlement measurements for all possible soil stratification composition and load combinations into features by the feature extraction method, developed in chapter 6. Thirdly, section 7.3 illustrates the choice of the target function and subsequently section 7.4 explains the learning model and the associated components. Lastly, in section 7.5 the training results are analysed and the quality and the capacity of the learning model is mapped.

## 7.1. Required Information

The environment is the collection of all possible features that can statistically be expected within the simulated heightening case. Hence, in order to construct this environment, the WBI-SOS subsoil scenarios with their layer interface distributions in combination with all possible load combinations are required. However, in subsection 6.2.2, it became clear that prior to applying a Machine Learning application the subsoil scenarios B and F could be distinguished from surface settlement measurements. The result is that the Machine Learning application only has to distinguish the remaining four subsoil scenarios. In conclusion, the required information from the simulated heightening case consist of the WBI-SOS subsoil scenarios A, C, D and E, including their layer interface distributions, in combination with the distributed load information. An overview of this data is shown in Figure 7.1.

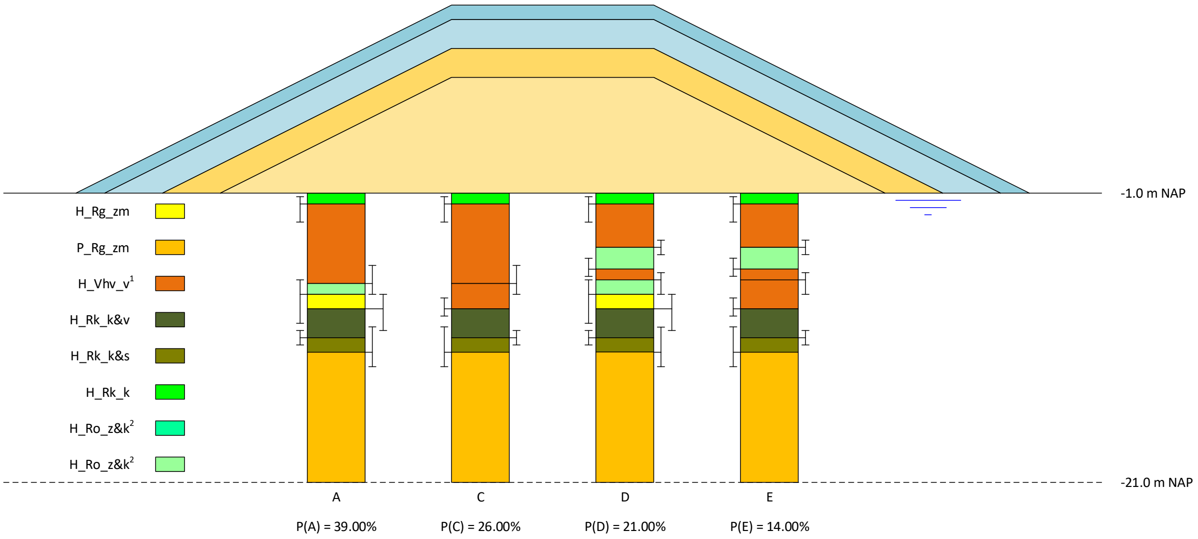


Figure 7.1: Visualisation of the required information from the simulated heightening case.

## 7.2. Constructing the Environment

In order to determine the expected prediction accuracy of the SVM, the environment is created. The environment includes all possible feature vectors that are to be expected in the dike segment, or in other words, it represent the expected input space.[1] This collection of feature vectors is created by calculating the synthetic surface settlement measurements for all possible combinations of loading curves and soil stratification compositions and transform them with the feature extraction method. The possible load curves within the dike segment are summarised in Figure 6.5b and all possible stratification compositions are summarised by the statistical overview formed by the WBI-SOS subsoil scenarios A, C, D and E. If the environment is constructed correctly, it represents the expected variation of soil stratification of the subsoil below the dike segment. Since no calibration to any measurement is conducted in this thesis, it is assumed that the created environment in this chapter does represent the total variation of soil stratification of the subsoil of WBI-SOS segment 15022.

### 7.2.1. Sampling Method

To obtain a representative approximation of the environment, per subsoil scenario 10 000 surface settlement measurements are calculated. In this subsection the sampling method is described that is used to obtain the required input for the D-settlement calculations, namely: the load combinations and the soil columns.

#### Sampling the Load Combinations

The 10 000 load combinations are randomly sampled from the 30 load combinations that have been derived in subsection 6.2.1. The 30 load combinations obtained from Figure 6.5b, where the cumulative load schematisation, shown in Figure 6.2b, are subdivided in time-dependent parts. Hence, each combination consists of a preload (which can be zero) and a new load component.

#### Sampling the Soil Columns

The 10 000 soil columns are sampled per subsoil scenario. The sampling procedure is visualised in Figure 7.2 and is conducted as following:

1. Given the median, the 10th and the 90th percentiles for each layer interface distribution per subsoil scenario, continuous triangular probability distributions are calculated for all layer interfaces.[2]
2. Each continuous distribution is approximated by a step-wise non-continuous distribution, to take into account the minimum resolution of 0.25 m of the subsoil scenarios.
3. Per layer interface 10 000 samples are drawn from the approximated step-wise non-continuous distribution.
4. Given the sampled layer depths, layer thicknesses are calculated and the soil columns are composed.

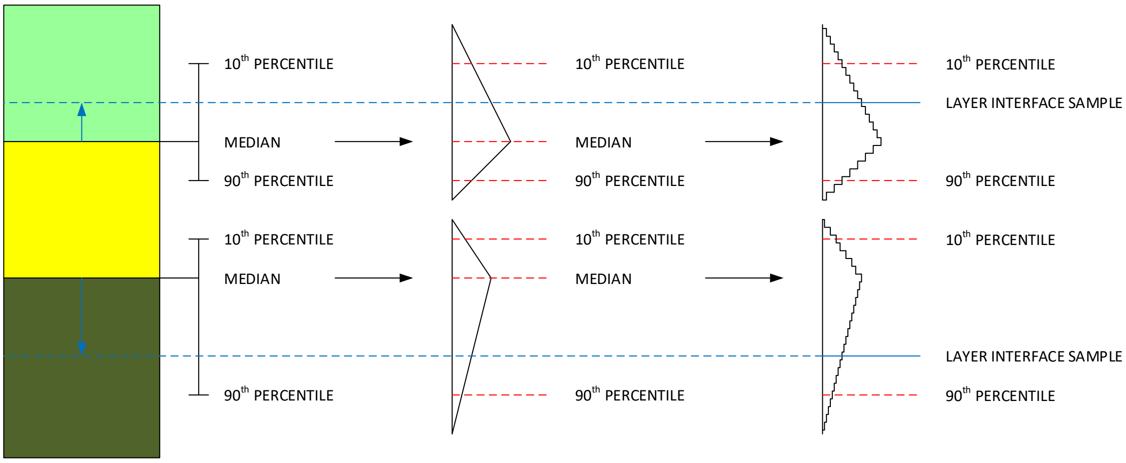


Figure 7.2: Visualisation of the layer interface sampling procedure.

[1] In a real application one has to calibrate the environment to local measured information.

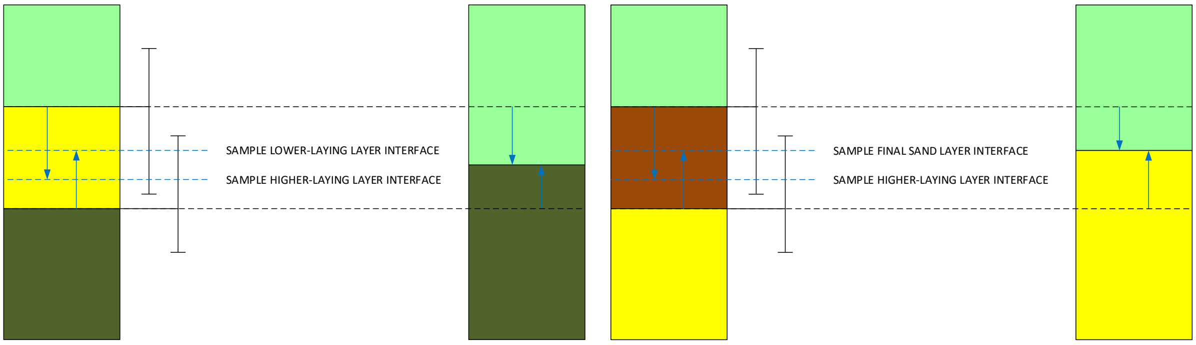
[2] Since only the values of median, the 10th and the 90th percentiles of each layer interface distribution are known, there is no information about the shape of the distribution. In accordance to "*Triangular Distributions* by Petty and Dye [35], given the available information, a continuous distribution should be obtained by using a triangular probability distribution.

### Sampling Rules

Due to the definition of the WBI-SOS subsoil scenarios, in multiple cases the layer interface distributions are overlapping each other.[3] Consequently, a layer with a negative thickness is created, which is obviously impossible in reality. Therefore the following sampling rules are applied:[4]

1. In case the depth of the a lower-laying layer interface is sampled above the depth of the higher-laying layer interface, the soil layer with the negative thickness disappears and both the adjacent layers equally compensate for the overlap.
2. In case the depth of the final sand layer interface is sampled above the depth of the higher-laying layer interface(s), the soil layer(s) with the negative thickness(es) disappear(s), the final sand layer interface is set to its sampled depth and the higher-laying layer interface equalises that depth.[5]

In Figure 7.3 both sampling rules have been visualised.


SAMPLE LOWER-LAYING LAYER INTERFACE
SAMPLE HIGHER-LAYING LAYER INTERFACE
SAMPLE FINAL SAND LAYER INTERFACE
SAMPLE HIGHER-LAYING LAYER INTERFACE

(a) Visualisation of sampling rule 1.

(b) Visualisation of sampling rule 2.

Figure 7.3: Visualisation of the two sampling rules in two possible cases of layer interface sampling overlap.

### Sampling Filters

After finalising the sampling procedure, to each sampled soil column a load combination sample is assigned. However, before the settlement curves can be calculated the samples have to be filtered. Filtering the samples is required for two reasons. The first is to maintain scenario uniqueness. Due to the usage of the sampling rules, soil layers can disappear during the sampling process of the layer interfaces. The outcome can become a soil column composition that becomes part of another subsoil scenario. These compositions have to be removed as they offset the scenario uniqueness and thereby making learning impossible. The second is to prevent the existence of duplicated combinations of soil column samples and load combination samples. Due to spatial variability of the subsoil and the composition of the dike body, it is unlikely to observe duplicates in reality. Therefore, the following sampling filters are applied:[6]

1. In case a composed soil column sample from any subsoil scenario can be identified as a soil column sample of the other subsoil scenarios, the soil column sample is removed.
2. In case a duplicate of a combination of a soil column sample and a load combination sample is created during the assigning process, the duplicate is removed.

[3] Note, the subsoil scenarios have not been calibrated to local measured stratification information and the targeted primary failure mechanism, therefore these overlaps exists. After calibration the subsoil scenarios should either not contain overlap in the layer interface distributions anymore.

[4] Due to these sampling rules, the statistics of the subsoil presented by each subsoil scenario is slightly disturbed. The effects of these distortions on the training results are discussed in subsection 7.5.2.

[5] The reason behind this rule is that two subsequent negative layer thicknesses can only occur if the final sand layer interface sample is involved. Numerically the correct position of the layer interface can be calculated through sampling rule 1, however the required accuracy with respect to the definition of the subsoil scenarios (vertical data resolution of 0.25 m) and the maximum allowable number of decimals in D-Settlement make this action meaningless. In combination with the low influence of the exact position of the lower boundary on the settlement curves (the influence of the load decreases with the depth), it is chosen to simply the formula that decides the position of the lower boundary.

[6] Similar to the sampling rules, these sampling filters disturb the statistics of the subsoil scenarios. Due to their usage, the probability of occurrence of the subsoil scenarios within the environment is slightly affected. However, since the training data is drawn with the original distribution of the probability of occurrences of the subsoil scenarios from the environment, the learning process is unaffected. Furthermore, as the error measure used in the test process of reconstructing the environment (see subsection 7.5.3) is independent of the label distribution of the environment (see explanation of the the Matthews Correlation Coefficient in subsection 7.4.4), the presented bias between the estimated and the measured prediction accuracy is neither influenced.

### 7.2.2. The Environment

Finally, the settlement curves are calculated by D-Settlement and subsequently transformed into features by the feature extraction method to create the environment. To clarify, the total process to obtain the environments included the following steps:

1. Sampling of 10 000 soil column samples per subsoil scenario, which are assigned each with a load combination sample.
2. Application of sampling filters to maintain scenario uniqueness and to remove undesired duplicates.
3. Calculation of settlement curves by D-Settlement given the combinations of soil column and load combination samples.
4. Extraction of features per settlement curve by the feature extraction method.

The obtained features form together the environment, which is presented in the two dimensional feature space in Figure 7.4. Per subsoil scenario, the population of feature vectors represents the variation of the soil stratification as defined by the layer interface distributions. In the same figure also the fit values of parameters *a* and *b* per subsoil scenario are shown, which have been derived in subsection 6.3. These fit values are the feature vectors that belong to the soil columns of the subsoil scenarios without layer interface distributions. Therefore, they indicate the theoretical mean of the associated feature vector population per subsoil scenario. It can be seen that the overlap between the population is tremendous in case the full layer stratification is worked out. This implies that the complexity of the expected input space in terms of WBI-SOS subsoil scenarios is too high to be represented by only two features.[7]

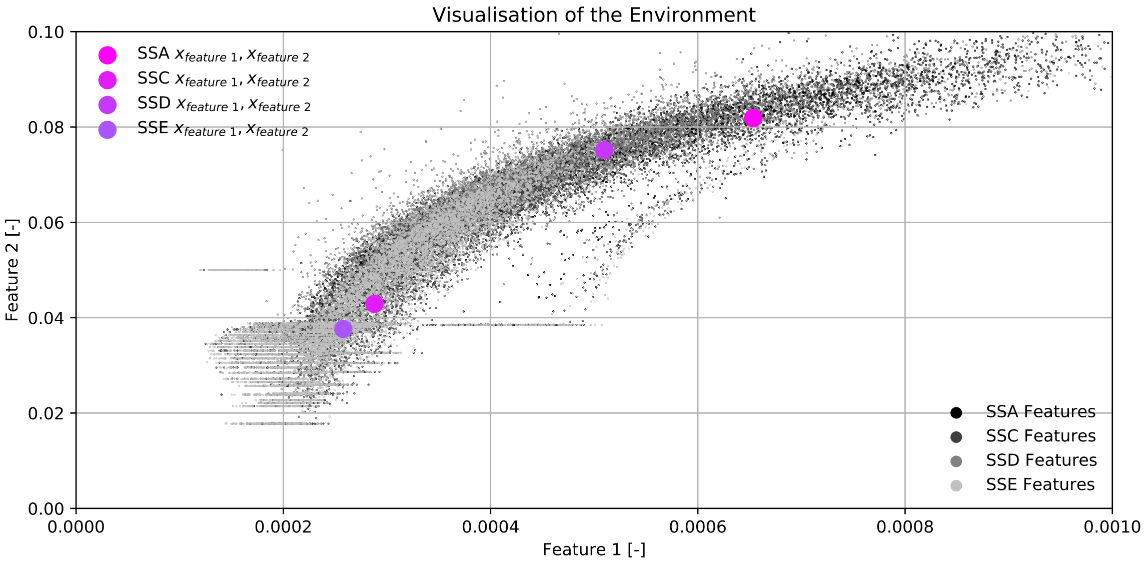


Figure 7.4: The constructed environment plotted in the feature space.[8]

[7] As explained in section 10.2, the focus of this thesis lays in the description of the Machine Learning processes, the associated techniques and the possible implementation of the expected outcomes. Therefore, it was decided to limit the feature space to only two dimensions. The consequences of this focus are encountered here, where a three-dimension feature space (which requires a minimum of three features) would have been the solution to obtain clearly distinguishable populations.

[8] The striking horizontal patterns that are developed by the features of subsoil scenarios C and E are the result of a limited fit freedom (two parameters) of the Power Model (see subsection 6.2.2) in combination with the decimal round up of both the input and the output of D-Settlement and the numerical errors the program produces for small load cases.

## 7.3. Defining the Target Function

As explained in subsection 5.3.1, the target function of the SVM describes the task of the algorithm to transform the input space into the preferred output space. The target function of the algorithm as described in subsection 2.1 is to produce an output space in terms of WBI-SOS subsoil scenarios. Unfortunately, an algorithm is not simply capable of performing any preferred task. In order to obtain a target function with a reasonable prediction accuracy, the target complexity of the input space should be matched with the quantity of the training data, to be certain that the features are capable of reflecting the target complexity of the input space and provides a sufficient test set. Hence, in this section first the preferred target complexity of the input space and the quantity of the training data are analysed in respectively 7.3.1 and 7.3.2. Secondly, in subsection 7.3.3, the target complexity of the input space is matched to the quantity of the training data.

### 7.3.1. The Preferred Target Complexity of the Input Space

For a classification task, the target complexity level of the input space depends on the ability of the feature vectors to represent the different classes that have to be distinguished from each other. According to Figure 7.4, the overlap of the feature vector populations within the environment is enormous. Therefore it can be concluded that the two features that are generated from the input space by the feature extraction method are not capable of representing the complexity of the preferred output space in terms of WBI-SOS subsoil scenarios. To acquire a Machine Learning algorithm that has a good prediction accuracy, either the number of features that are extracted by feature extraction method have to increase or the target complexity of in the input space has to decrease.

### 7.3.2. The Available Quantity of the Training Data

The average quantity of available CPT and borehole measurements within a dike segment can be expected to be of tens up to maximum a couple of hundred. However, prior to the training phase, it is hard to to tell if the quantity of the training data is sufficient to reflect the target complexity of the input space. It depends strongly on the number of features, their correlation and the label distribution. According to Beleites et al. [7], in case of multi-classification more important is a sufficient quantity of the test set rather than reflecting the target complexity of the input space in the training set. The authors mention that the quantity of the training set can be determined from learning curves, which show the expected $E_{out}(g)$ and the measured $E_{in}(g)$ as a function of the quantity of the training data. For testing, the authors recommend at least 75 data points per class, since fewer numbers cannot deliver reliable learning curves anymore in case of multi-classification.

This indicates that the order of the available quantity in reality will most likely provide the sufficient quantities for both the training and test set. If not, extra local measurements can be conducted to acquire sufficient training data if measurement costs are outweighed by potential expected benefits.

### 7.3.3. Matching the Target Complexity to the Feature Representativity of the Input Space

In the previous subsection it has been argued that the quantity of available training data is not an issue. The preferred target complexity of the input space, however, cannot be realised given the two features that are extracted by the feature extraction method. Hence, either the target complexity of the input space needs to reduced or the number of features need to increase. The latter interferes with the set limit of a two-dimensional feature space, therefore in this subsection a reduced target complexity is tested.

From the conclusions of Figure 6.12, it became clear that classes of subsoil scenarios were distinguishable. Class 1 and class 2 are formed respectively by the subsoil scenarios A, D and C, E. The use of such classes would reduce the complexity of the target function into a simple binary classification problem. Then the question rises if such drastic simplification will still provide any useful information. Taking a clear look into Figure 7.1, it shows that class 1 differentiates from class 2 by containing a sand layer. For example, in case one is interested into the detection of cross-sectional corridors prone to the primary failure mechanism piping, knowing where to expect soil columns that include a sand layer would already be sufficient. Therefore, it can be stated that the simplification into detecting two classes is still useful.

The result of the classified environment is shown in Figure 7.5. Although the overlap between the classes is still sufficient, which means the prediction accuracy will include a certain error rate, clearly two populations can be distinguished. In conclusion, the reduction of the target complexity of the input space from WBI-SOS subsoil scenarios into two distinguishable classes of WBI-SOS subsoil scenarios provides a proper match with the available quantity of training data.

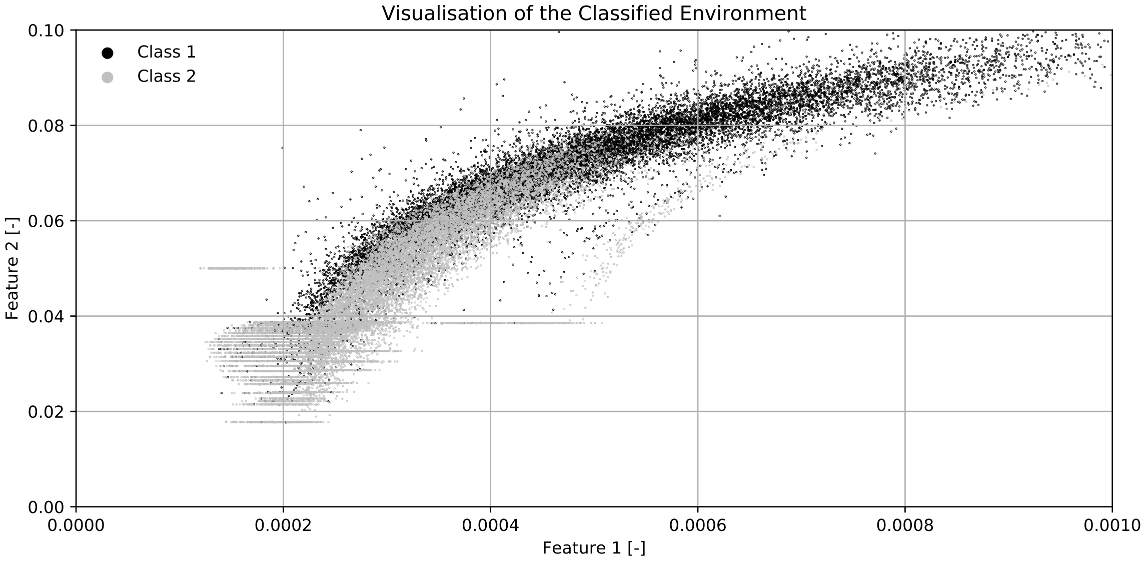


Figure 7.5: The classified environment plotted in the feature space.

## 7.4. The Processes of the Learning Process

In order to achieve an unbiased final hypothesis, it is important to prevent any accidental leakage of knowledge into the learning process. Therefore, in general one makes use of a learning model. In this section the composition of the learning model is explained. First, in section 7.4.1 the required input and the tasks of the learning model are summarised and an overview of the learning process conducted by the learning model is shown. Subsequently, in the subsections 7.4.2, 7.4.3 and 7.4.4 each task of the learning model is discussed in detail.

### 7.4.1. The Learning Model

The required input of the learning model are the training data and the hypothesis set. The training data is acquired by taking random samples of specific sizes from the constructed classified environment. This sampling procedure takes the probability of occurrence of each WBI-SOS subsoil scenario into account to ensure statistical equality with the WBI-SOS subsoil scenarios.[9] As defined in subsection 5.4.1, the hypothesis set consist of a SVM, equipped with a Gaussian Radial Basis Function, that is tested for varying hyper-parameters. The hyper-parameters and their test ranges are explained in subsection 7.4.3.

Given this input, the learning model automates the learning process and provides a final hypothesis *f*. This automated process conducts the following three tasks:

- Pre-processing the training data.
- Performing a grid search cross-validation to find the best hypothesis with the Bias-Variance Analysis.
- Measuring the performance from the training metrics.

In Figure 7.6 an overview is shown of the learning process that is conducted by the learning model in this thesis. It can be seen that structure is equal to the learning process shown in Figure 5.8, where a correctly performed learning process of multiple hypothesis is shown that assures no leakage of knowledge into the learning process. Hence, it can be expected that the constructed learning model provides an unbiased performance measure. For the construction of the learning model the open source Python Machine Learning library *scikit-learn* [2] is used. The functions of these libraries are presented by their names in Figure 7.6. Their purpose and tasks will be explained in the following subsections.

[9] In order to sample statistically equal training data, the sampled training data should also represent the equal layer interface distributions. This is not controlled in this thesis. In subsection 7.5.2 the possible effects are mapped.

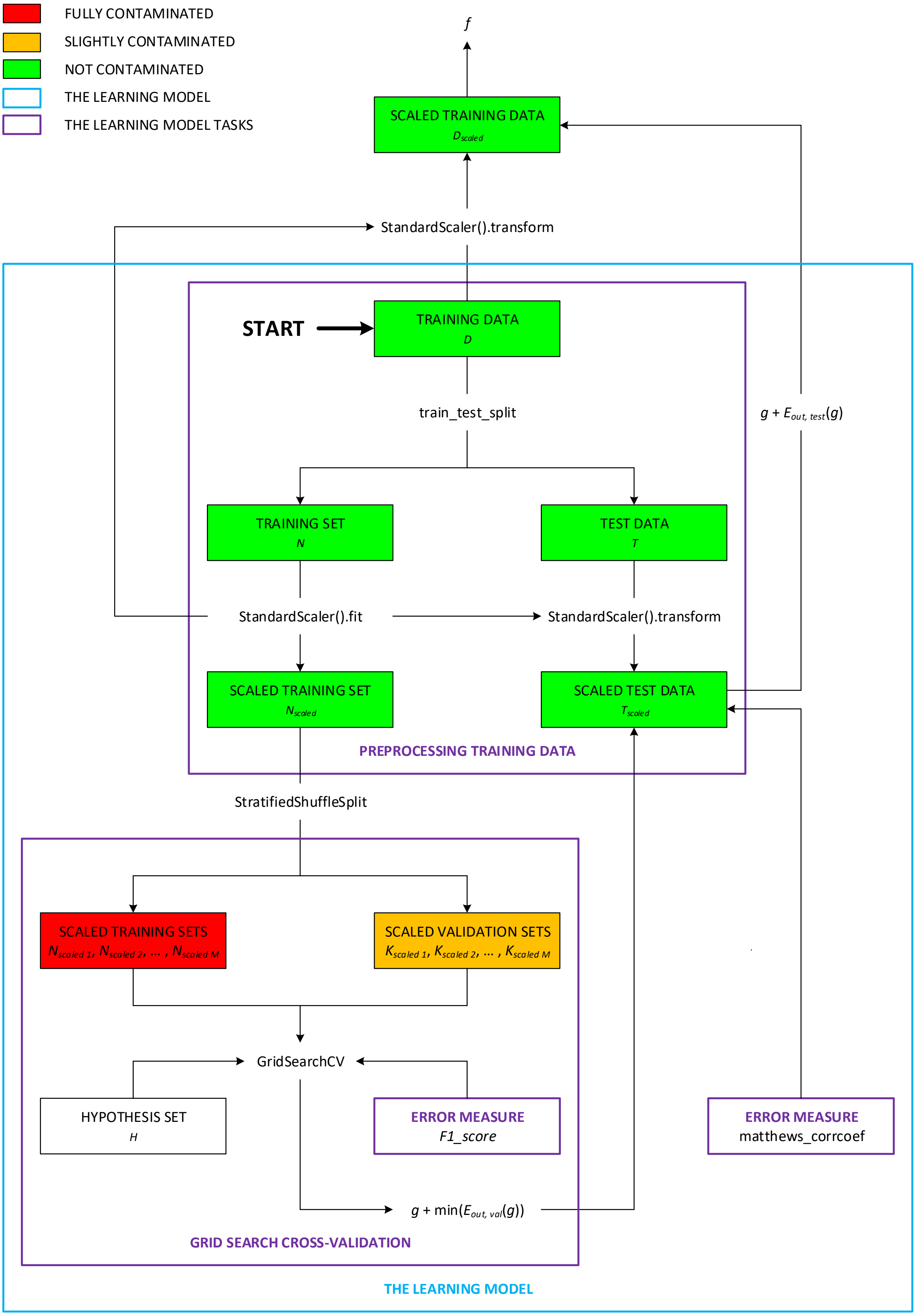


Figure 7.6: The learning process and its components.

### 7.4.2. Pre-processing the Training Data

The first task of the learning model is to pre-process the training data. This includes splitting of the training data into a training and a test set and subsequently scaling the features of both sets. In Figure 7.6 the box that marks pre-processing the training data visualises the context that will be discussed in this subsection.

First, the training data is split randomly with the ratio 75% and 25% into respectively a training and test set by the function *train_test_split*. This function has the attribute that it randomly splits the training data, however maintains the label distribution as sampled from the classified environment. Therefore, both training and test sets will remain statistically equal to the training data with respect to the probability of occurrence of each WBI-SOS subsoil scenario.

After splitting the training data, the features have to be scaled. Scaling is of high importance in case of using a SVM. According to Raschka [36], feature scaling can be conducted through normalisation and standardisation. Although normalisation is usefull, standardisation can be more practical. With standardisation the features are centred at a mean = 0 with a standard deviation of 1. Due to the initial weight values of a SVM, which are zero or near zero, it becomes more easily to learn the weights. Another advantage of standardisation over normalisation is that is maintains information regarding outliers.

The scaling is performed by the function *StandardScaler().fit*. Important to mention, the scaler is fit only once to the training set. Any data point that is later used for both testing and predicting matters, has to be scaled accordingly. This is done by using the function *StandardScaler().transform*.

### 7.4.3. Performing a Grid Search Cross-Validation

Raschka [36] mentions that in Supervised Learning two types of parameters are learned: those that are learned from the training data, such as the weights of the SVM, and the tuning parameters, also called hyper-parameters, of which the regularisation parameter is an example. A grid search cross-validation is a tool that can be used to find the optimal parameters combination of both types, which uses a cross-validation strategy to create multiple training and validation sets and, subsequently, tests all possible combinations of hyper-parameters on them. Then, by means of the results of an error measure the best hypothesis (the one that contains the optimal parameter combination) can be selected to function as the final hypothesis. To perform the grid search cross-validation the function *GridSearchCV* is used. To find the best hypothesis, the performance of each hypothesis in terms is measured by the error measure $F_1$ Score.

First, the scaled training set is split with a cross-validation iterator into multiple folds by a certain cross-validation strategy. The strategy used in this thesis is called the StratifiedShuffleSplit, which is executed by using the equally named function *StratifiedShuffleSplit*. This cross-validation iterator randomly splits the scaled training set into 10 folds, of which each contains a scaled training and validation set of respectively 75% and 25% in proportion to the fold size. During the splitting procedure, similar like *train_test_split*, the label distribution of the training data is maintained in order to ensure statistical equality.

Secondly, as mentioned in the introduction of this subsection, the hypothesis set consists of a SVM, equipped with a Gaussian Radial Basis Function kernel, that is tested for varying hyper-parameters.[10] The hyper-parameters C is the inversed regularisation parameter $\lambda$, related to the soft margin slack $\xi$.[11] This parameter controls the ability to capture the noise of the training data, or in other words, it controls the area of overlap between the populations of both classes that is taken into account in the calculation of the optimal decision plane. The larger C, the smaller the allowable slack (the smaller the area of overlap) and vice versa.[12]

[10] The choice for and the concept of a Support Vector Machine and a Gaussian Radial Basis Function kernel is explained in section 5.4.

[11] The purpose of the regularisation parameter $\lambda$ and the concept of the soft margin slack $\xi$ are explained in respectively subsection 5.3.5 and 5.4.1.

[12] In order to clarify the relation between C and $\xi$, a small example is given on the cases presented by Figure 5.3, where the concept of underfitting and overfitting a classification problem with respect to an appropriate fit is illustrated. In the left figure it can be seen that the area of the overlap around the decision plane was too small (the noise level of the training data was underestimated). This means that C was chosen too large and thus only a few outliers were taken into account in the calculation of the decision plane ($\xi$ was too small). Consequently, the real structure of the data was not captured and therefore underfitting occured. In the right figure the area was too large, which means C was chosen too small (the noise level of the training data was overestimated). Too many outliers were taken into account in the calculation of the decision plane ($\xi$ was too large). Consequently, the noise level was seen as the data structure, with overfitting as the result.

The hyper-parameter $\gamma$ is the Gaussian Radial Basis Function kernel parameter, which controls the influence of a single support vector in correctly classifying any training data point during the training process. In other words, it controls the relative contribution of the Bias with respect to the Bias-Variance Trade-off.[13] The larger $\gamma$, the lesser the influence range of a single support vector, and thus the larger the freedom of the model to capture the structure of the training data.[14] This implies that the Bias, the structural differences between the final hypothesis and unknown target function, becomes smaller but the Variance, the variable error for any given prediction, becomes higher. Vice versa, if $\gamma$ is small, the influence of a single support vector on classifying any training data point is high. This reduces the risk of learning the training data, which results in a high Bias - low Variance model.

In Hsu et al. [22], the hypothesis set is optimised to reduce the exhaustive search of testing the full hypothesis set. This is done by firstly calculating the parameter combination that provides the highest prediction accuracy in a coarse grid with exponential growing sequences C and $\gamma$ values. The obtained combination serves as an initial guess to estimate the region of promising parameter combinations. Subsequently, a finer grid that covers that specific region is used to find the optimal parameter combination for the final hypothesis. In this thesis a similar approach is used.[15] At first a coarse grid of C = $1{\cdot}10^{-2}$, $2.15{\cdot}10^{-1}$, ..., $1{\cdot}10^{10}$ and $\gamma$ = $1{\cdot}10^{-9}$, $2.15{\cdot}10^{-8}$, ... , $1{\cdot}10^{3}$ is used.[16] The best combination found in this search will serve as the center of the finer linear grid that delivers the optimal combination, spans from 0.1 · center to 10 · center. Figure 7.7 visualises an example of the application of the coarse logarithmic grid, of which the most promising region is analysed by a finer linear grid.

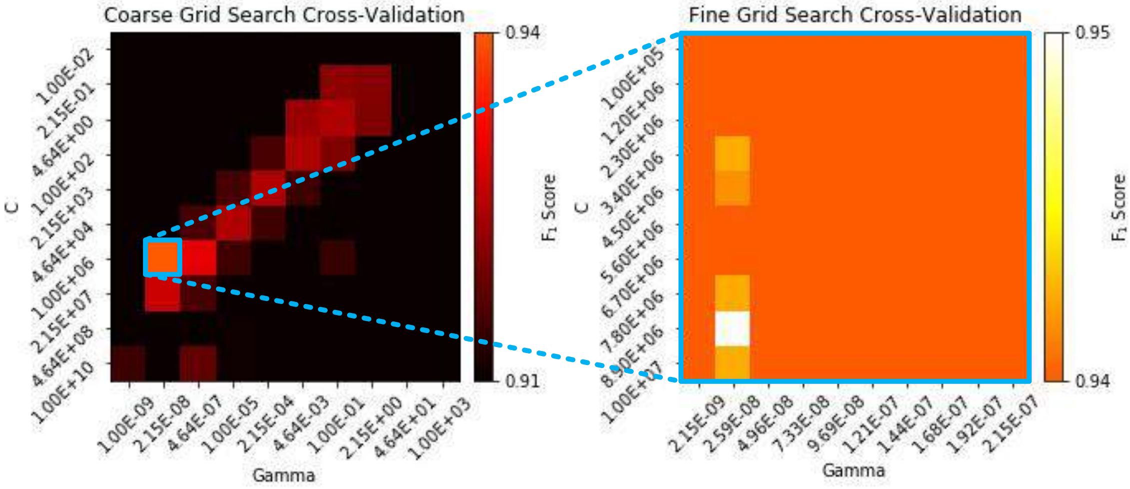


Figure 7.7: Example of using a coarse grid and a fine grid in order to find the optimal hyper-parameter combination.

The coloured areas of the left plot show the top 20% best promising regions, others are coloured in black. The coloured areas in the right plot, show the parameter combinations which result in better prediction accuracies with respect to the prediction accuracy of the best parameter combination found by searching the coarse grid. Combinations in the fine grid which represent parameter combinations with lower performances compared to the best estimated parameter combination found in the coarse grid, are coloured orange.

[13] The concepts of the Bias and the Variance and their inevitable relation (Bias-Variance Trade-off) are explained in subsection 5.3.3.

[14] If $\gamma$ is chosen to large, the model has too much freedom and becomes prone to overfitting.

[15] Note, the reduction of the exhaustive search has also a disadvantage. Since the hyper-parameters have the ability to enhance and weaken each others influence, certain parameter combinations can lead to almost or completely equal performance. Therefore, there is always a probability that the promising area that is selected from the coarse grid does not contain the most optimal parameter combination.

[16] The ranges per parameter contain each 9 intervals between the minimum and maximum values on a logarithmic scale with base 10.

### 7.4.4. Measuring the Performance of the Learning Process

Since the target function of the algorithm is set to perform binary classification, see subsection 7.3.3, according to Chicco [14] the best error measure to use is the Matthews Correlation Coefficient. Unfortunately, *GridSearchCV* did not include this error measure by default. Although they support tailored solutions, due to time constrains it was not possible to implement such alternative. Therefore, the second best method for binary classification, the $F_1$ Score, has been used during the grid search cross-validation in order to measure the performance of each hypothesis being tested.[17] The Matthews Correlation Coefficient is used to estimate the real $E_{out}(g)$ of the final hypothesis during the performance measure of the best hypothesis on the test set. In this subsection it is explained how both methods measure the performance and how their output is related to Bias-Variance Trade-off.

In the explanation of both error measure methods, typically the following abbreviations are used:

$TP$ = Correctly predicted positive cases (True Positive)
$TN$ = Correctly predicted negative cases (True Negative
$FP$ = Incorrectly predicted positive cases (False Positive)
$FN$ = Incorrectly predicted negative cases (False Negative)

#### $F_1$ Score

The $F_1$ Score measures the accuracy of a binary classifier through the harmonic mean of the measured Precision and Recall per label. Precision and Recall are measured according to respectively Equation 7.1 and Eguation 7.2.

$$Precision = \frac{TP}{TP+FP} \tag{7.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{7.2}$$

The harmonic mean of both is found according to Equation 7.3:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{7.3}$$

In relation to the Bias-Variance Trade-off, a high Precision and a low Recall corresponds to the case of having a high Bias and a low Variance and visa versa. Therefore, a good model contains a high Precision and a high Recall. The $F_1$ Score shows the average of both on a scale from 0 to 1, where 0 means both a low Precision and Recall and 1 means both a high Precision and Recall.

#### Matthews Correlation Coefficient

Boughorbel et al. [12] state that the Matthews Correlation Coefficient (MCC) takes true and false positives and negatives into account and is generally regarded as a balanced measure which can be used even if the class distribution is skewed. In comparison to the $F_1$ Score, this error measure is not affected by unbalanced training data samples. The MCC measures the performance according to Equation 7.4.

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}} \tag{7.4}$$

The relation to the Bias-Variance Trade-off cannot clearly be explained. The MCC returns an output between -1 and 1. In case of -1, it means that the classifier predicts the classes 100% incorrect. In contrast, in case of 1 the classifier predicts the classes 100% correct. If the output is zero, it means a random classifier has been trained.

[17] The effects of using this error measure is discussed in 9.2.4.

## 7.5. The Training Results

With the target function defined and the learning model constructed, in this section the performance of the Machine Learning application is studied. First, in subsection 7.5.1, the output of the learning process is shown and explained. Secondly, the influence of the random seed in sampling and splitting the training data is quantified and investigated in subsection 7.5.2 and lastly, in subsection 7.5.3 the influence on the prediction accuracy by the quantity of training data and the target complexity of the input space is mapped.

### 7.5.1. Output of the Learning Process

Given any training data sample originating from the classified environment derived in subsection 7.3.3, during the learning process, which is executed by the learning model (described in section 7.4), each hypothesis from the hypothesis set (described in subsection 7.4.3) is tested and finally a trained final hypothesis is obtained. In this subsection the results of the learning process of a final hypothesis, trained on a certain training data sample, is visualised and discussed.

The training data sample used to produce the final hypothesis contains 100 labelled training data points and originates from the classified environment, which is sampled with 25-75 sample bounds.[18] The first output that is generated by the learning model is a visualisation of the grid search cross-validation, visualised in Figure 7.8. In the left plot the search through the coarse grid has yield a estimated region of promising hyper-parameter combinations, which is shown in right plot. It can be seen in the right plot that only one combination resulted in a higher $F_1$ Score, which implies that the hyper-parameter combination ($C = 2.15{\cdot}10^2$ and $\gamma = 4.64{\cdot}10^{-2}$) is the best hypothesis to propose for the final hypothesis.

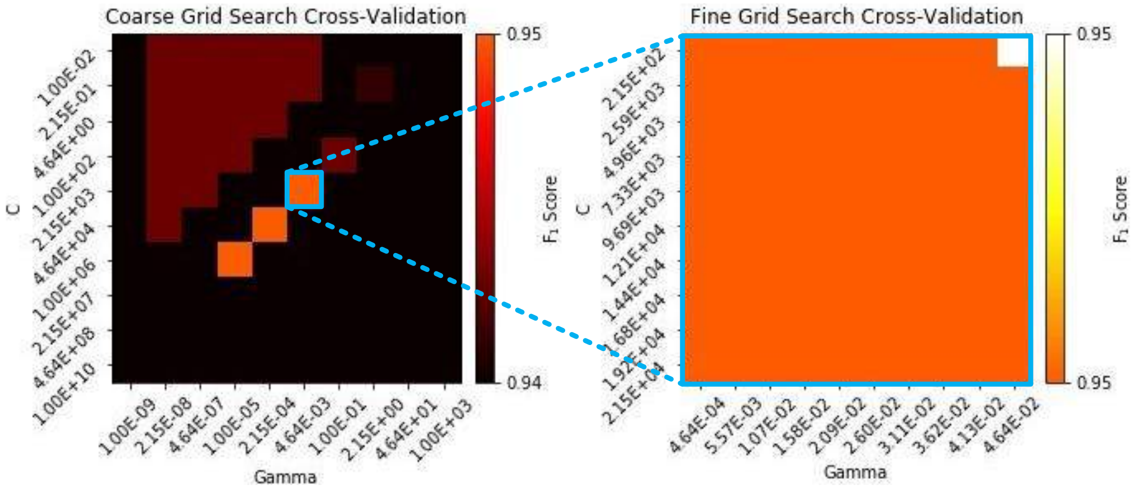


Figure 7.8: Visualisation of the grid search cross-validation results.[19]

In Table 7.1 the metrics of the learning process are shown, including the prediction performance measured while predicting the full classified environment, from which the training data has been sampled.

Table 7.1: Training performance on 100 labelled data points.

| | **C** [-] | $\gamma$ [-] | **$F_1$ Score** [-] | **MCC** [-] |
|---|---|---|---|---|
| Coarse grid | $2.15{\cdot}10^3$ | $4.64{\cdot}10^{-3}$ | 0.95 | - |
| Finer grid | $2.15{\cdot}10^2$ | $4.64{\cdot}10^{-2}$ | 0.95 | - |
| Test set | - | - | - | 0.85 |
| Classified environment | - | - | - | 0.87 |

[18] This means that only soil column samples could be drawn from limited soil layer distributions (between the 25th and the 75th percentiles). In Figure 7.12 the functioning of the sample bounds is illustrated.

[19] The difference between the prediction accuracy of the optimal hyper-parameter combination found in the fine grid ($F_1$ Score = 0.947) and the promising combination found is coarse grid ($F_1$ Score = 0.948) is minuscule. Taking into account the disadvantages of the $F_1$ Score (see subsection 7.4.4), it becomes questionable if the found combination is the most optimal combination. However, subsection 7.4.3 explains that certain hyper-parameter combinations lead to almost or completely equal performance due to their ability to enhance and weaken each others influence. Given the differences between the performance in combination with the quality of the performance measure, it can be concluded that as long as the hyper-parameters are chosen from these three promising regions, it is assured that the highest qualified classifier that could be trained on the available training data sample is obtained.

From these metrics an important conclusion about the quality of the learning process can be drawn. The performance on the test set (MCC = 0.85) is not equal to the performance on the classified environment (MCC = 0.87). This implies that the estimated $E_{out}(g)$ (1 - 0.85 = 0.15) measured on the test set, does not properly reflect the real $E_{out}(g)$ (1 - 0.87 = 0.13), which is measured on the classified environment. Note, this difference (the bias) in reality is a little smaller than reflected in this comparison, since the final model is trained on all available training data before its performance is being tested on the classified environment.[20] Subsection 7.5.2 quantifies and explains the phenomenon that causes this bias between the estimated and the real $E_{out}(g)$.[21]

In Figure 7.9 the decision function of the final hypothesis is visualised in the input space. With this decision function is becomes possible to classify any data point from the input space. In black and grey the classified environment is shown. Through observation, it can be seen that class 1 and class 2 do overlap each other considerably. In blue and red the classified training data, sampled from the classified environment, is presented. Considerable overlap can also be seen in the population of both classes. The decision function, shown by the decision boundary, is based on the optimal margin. The optimal margin is the area between the two hyperplanes, which are supported by the support vectors, indicated by training data points that contain an aqua coloured ring. The background fill shows the normalised distance of the feature space with the respect to the decision plane, which can be used to indicate the transformation of the feature space into the space in which the optimal margin was found.

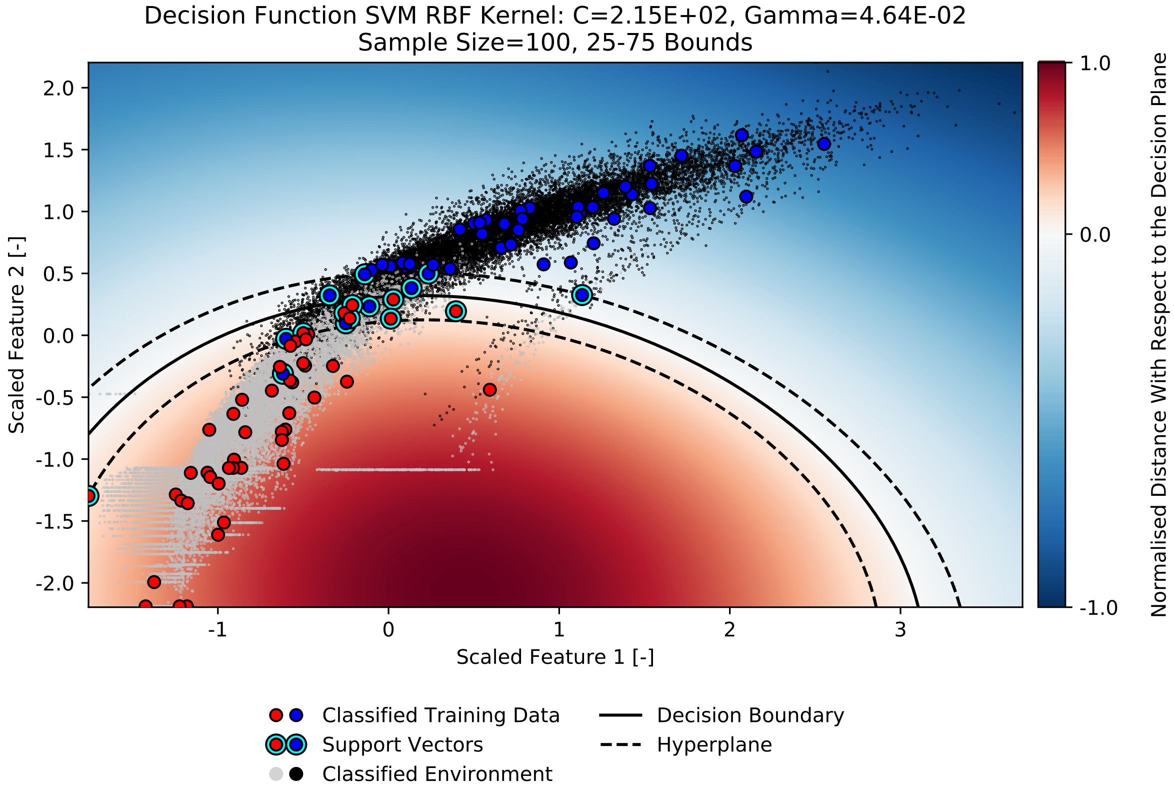


Figure 7.9: Visualisation of the decision plane of the final hypothesis within the feature space.

[20] After the final hypothesis is found, the final hypothesis is retrained on the all the training data points. Since the training process is only conducted on the training data set, this action increases the available training data points. The result is that training data reflects the classified environment better and therefore the final model will approach the estimated $E_{out}(g)$ more than shown by measured performance on the test.

[21] In subsection 5.3.2 the importance of a generalising model has been explained. Hence, it vital that the estimated $E_{out}(g)$ reflects the real $E_{out}(g)$. In this thesis the training is conducted on synthetic data, therefore also the output space (the labels of the classified environment) is known. Therefore, the bias between the estimated and the real $E_{out}(g)$ can be quantified and analysed. Important to realise is that in reality the output space is unknown, therefore the estimated $E_{out}(g)$ is the only indication of the real $E_{out}(g)$. It is only possible to approximate the bias through validation on newly acquired training data points.

### 7.5.2. Bias Due to the Random Seed

In the previous subsection a bias between the estimated and real $E_{out}(g)$ was observed, even though a correctly performed learning process has been conducted by the learning model. This bias can be addressed to two phenomena:

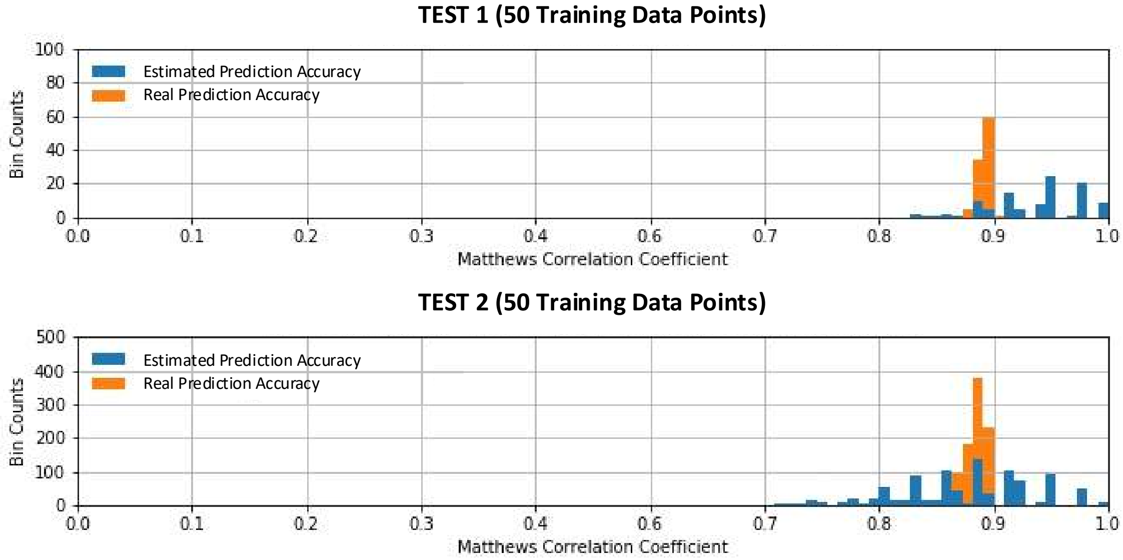
- The usage of a random seed in the sampling procedures of the training data.
- The usage of a random seed in the splitting procedures of the training data.

As explained in subsection 7.4.1, training data is acquired by taking random samples of specific sizes from a constructed classified environment. This sampling procedure assures that distribution of the classes is statistically equal to the distribution of the probability of occurrence of the WBI-SOS subsoil scenarios. However, it does not verify if the sampled training data also represent similar layer interface distributions as presented by the subsoil scenario. Consequently, the training data does not 100% reflects the classified environment. Exactly the same holds for each splitting of the sampled training data that is made during the grid search cross-validation. The splits that are automated by the functions *train_test_split* and *StratifiedShuffleSplit*, which only take into account that the distribution the classes in each duplication is statistical equal to distribution of the probability of occurrence of the WBI-SOS subsoil scenarios.

In this subsection the contribution of each phenomenon to the bias is analysed and quantified by conducting two tests. In test 1 the learning model repeats the training process 100 times on a single training data sample. In test 2 the learning model repeats the training process 10 times[22] on 100 training data samples. In each training session the found final hypothesis is tested on the classified environment to quantify the bias between the estimated $E_{out}(g)$ (measured during the training by testing the final hypothesis on the test set) and the real $E_{out}(g)$ (measured by testing of the final hypothesis on the classified environment). After conducting all the training sessions, the estimated and real prediction accuracies are plotted. By comparing the obtained distributions, it becomes clear which part of the bias (the difference between the distributions) is caused by which phenomenon. Since the estimated $E_{out}(g)$ depends on the quantity of the training data (see subsection 5.3.3), the bias depends similar on the quantity of the training data.[23] To analyse which phenomenon is the most sensitive to this dependency, the two tests are conducted on two training data sets of respectively 50 and 250 training data points.

#### Test Results 50 Training Data Points

Figure 7.10) presents the results of both the tests on 50 training data points. The distributions of the estimated $E_{out}(g)$ and the real $E_{out}(g)$ are shown respectively in blue and orange.


TEST 1 (50 Training Data Points)

TEST 2 (50 Training Data Points)

Figure 7.10: Test result distributions of test 1 (top plot) and test 2 (bottom plot) for a training session on 50 training data points.

[22] Due computational power constrains, the preferred 100 training repetitions on each training data sample could not be conducted.

[23] The more training data points, the better the data structure of the input space is reflected in the training data. Hence, a higher probability of learning the data structure of the input space instead of learning the data structure of the training data.

From test 1 it can be observed that the distribution of the estimated prediction accuracy is very irregular. It varies approximately between MCC = 0.21 and MCC = 1.0 and contains large empty spaces. In contrast, the distribution of the real prediction accuracy is smoother and varies between MCC = 0.79 and MCC = 0.88. By analysing the results of test 2, it can be seen that the distributions of both the estimated and the real prediction accuracy are similar to the ones observed in test 1. From these observations the following conclusions can be drawn:

- The distributions of both the estimated and the real prediction accuracy are quite similar for both tests. Since only test 2 included sampling of the multiple training data samples, it appears that the contribution to the bias by using the random seed in the sample procedure is inferior compared to the usage of the random seed in the splitting procedures.[24]

- In both tests, the distribution of the real prediction accuracy is significantly narrower compared to the distributions of the estimated prediction accuracy. This indicates that the size of the validation and test sets were not sufficient to provide a clear estimation of the estimated prediction accuracy (which explains the irregular distribution of the estimated prediction accuracy in both tests).[25] This implies that the algorithm is more robust than the error measure presents in cases of low training data quantities.[26]

### Test Results 250 Training Data Points

To test which phenomenon is the most sensitive to the quantity of the training data, both the tests have been repeated on a training data sample of 250 training data points. The results are presented by Figure 7.11. Again, the distributions of the estimated prediction accuracy and the real prediction accuracy are shown respectively in blue and orange.

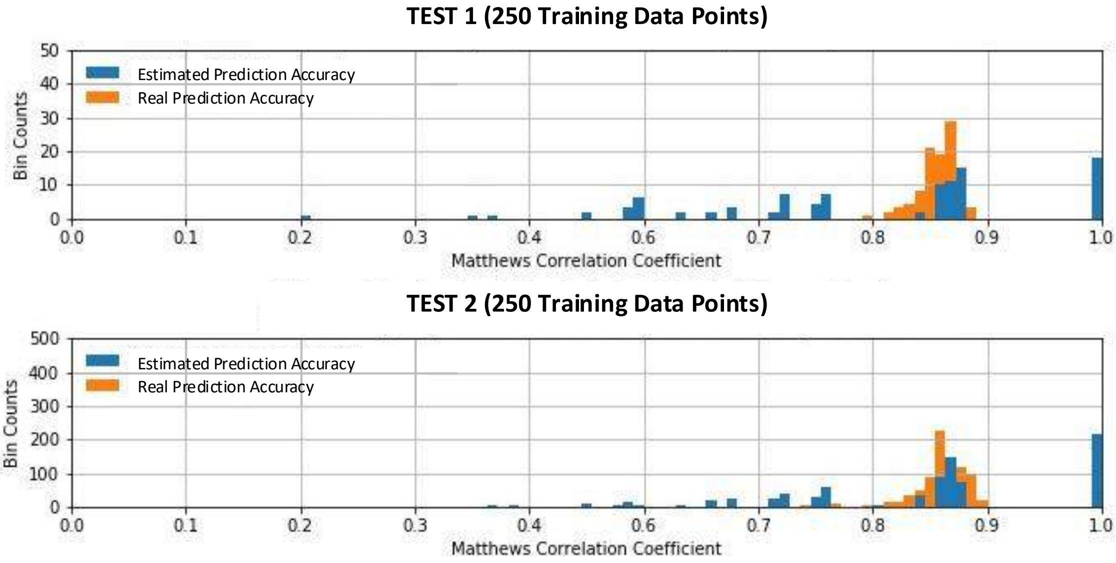


Figure 7.11: Test result distributions of test 1 (top plot) and test 2 (bottom plot) for a training session on 250 training data points.

In both tests it can be observed that the smoothness of the distribution of the estimated prediction accuracy improved significantly and it approximates the distribution of the estimated prediction accuracy more accurately (in test 1 it varies approximately between MCC = 0.83 and MCC = 1.0 and in test 2 between MCC = 0.72 and MCC = 1.0). However, in test 1 the distribution of the estimated prediction accuracy is positively shifted with respect to the distribution of the real prediction accuracy.[27], while in test 2 the distributions of the estimated and real prediction accuracy are not shifted. From these observations the following conclusions can be drawn:

[24] This makes sense since the sampling procedure is conducted only once, while the splitting procedure occurs multiple times.

[25] The low quantity of training data resulted in limited variation in the composition of the training, validation an test sets that were used for the training sessions. Consequently, the algorithm found often the same hyper-parameters for the final hypothesis.

[26] A Support Vector Machine decides the decision boundary based on the support vectors (see subsection 7.4.3). As long as training data sample is a spatially distributed sample, thereby representing roughly the distinguishable populations, the algorithm will find a good approximation of the decision boundary between the populations.

[27] With a positive shift is meant that higher performance scores are presented compared the real performance scores.

- Due to the increasing statistical equality of the training data with input space, the total bias decreased with respect to the tests performed on the 50 training data points (the estimated prediction accuracy) distributions approximated the real prediction accuracy more accurately).
- The clear positive shift with respect to the real prediction accuracy distribution indicates a stronger dependency to the drawn training data sample. This implies that the relative contribution to the bias by the usage of the random seed in the sampling process increased.
- The decreasing width and the increasing smoothness of the estimated prediction accuracy distribution in both tests indicates the relative contribution to the bias by the usage of the random seed in the splitting procedures decreased. However, the irregularities that are still present within the distributions show that the training data quantity is not yet sufficient to properly quantify the contribution.

### 7.5.3. Target Complexity of the Input Space Versus Quantity of the Training Data

In this subsection the influence on the prediction accuracy by the quantity of the training data and the target complexity level of the input space is mapped. The target complexity level of the input space is regulated by the usage of sample bounds, which constrain the available percentiles of the layer interface distributions from which a classified environment is sampled. In Figure 7.12 an example of such constrained distributions is shown. Associated to certain n - m sample bounds, the black areas in the triangular probability distributions represent the constrained percentiles from which no layer interface sample can be drawn. Note, once bounds are set, they are equally applied to all layer distributions corresponding to all WBI-SOS subsoil scenarios. The sample bounds are used to control the target complexity of the classified environment. For example, lower bounds (45-55) allow almost no layer interface variation, which results in nearly any overlap between class 1 and 2 in the feature space, while higher bounds (5-95) allow almost all layer interface variation, resulting in tremendous overlap.

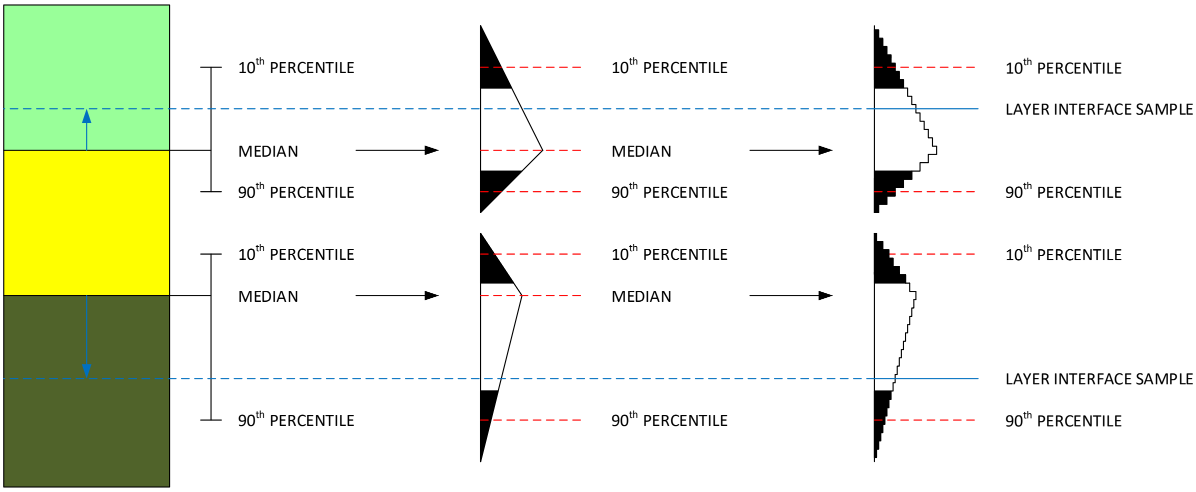


Figure 7.12: Example of the sample bounds constraining the layer interface distributions.

To test the influence of the target complexity of the input space, classified environments have been created for target complexity levels that gradually increase in steps of 5% from the 45th - 55th to the 0th - 100th percentiles. The latter represents the classified environment as shown in Figure 7.5. The ones sampled with lower sample bounds can be seen as more accurately defined subsoil scenarios, with respect to the classified environment associated to the 0-100 sample bounds. To test the influence of the training data quantity, per target complexity level 10 training data samples have been drawn from the associated classified environment, of which the size gradually increased in steps of 50 from 50 to 500 training data points. Figure 7.13 shows for each combination a 10 training repetitions averaged estimated predicting accuracy (left plot) and the absolute bias between the estimated and the real prediction accuracy (right plot) on the corresponding classified environments.[28] This average bias roughly indicates the standard deviation of the learning process.[29]

[28] Note, as mentioned before, since the output space (the labels of the classified environment) is unknown, in reality this bias cannot be quantified. The estimated prediction accuracy can only be validated.

[29] The phenomena that causing the bias are quantified and explained in subsection 7.5.2.

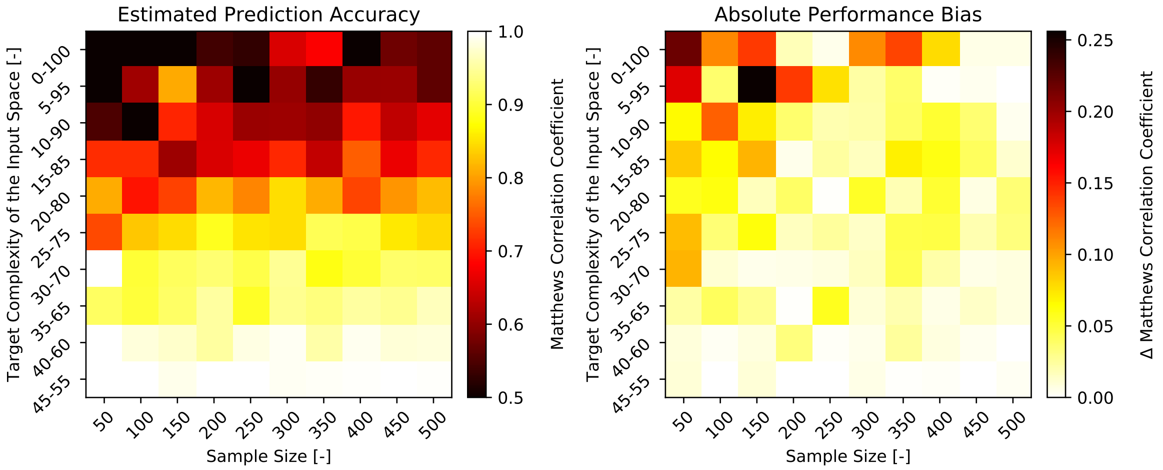


Figure 7.13: Influence on the estimated prediction accuracy by the quantity of the training data and the target complexity of the input space (left plot) and the corresponding absolute bias between the estimated and the real prediction accuracy (right plot).

For varying situations, Figure 7.13 presents a general performance estimation of the proof of concept that is presented in this thesis, including the associated standard deviation. From the left plot it can be seen that over the horizontal axis the quantity of training data used for the learning process increases, but the prediction accuracy barely increases. Over the vertical axis, as the target complexity of the input space increases and thus the overlap between the classes in the associated classified environments increases, the prediction accuracy decreases gradually from 100% (MCC = 1.0) to approximately 75% (MCC = 0.5). In the right plot the bias between the estimated and the real prediction accuracy is presented. It can be seen that the disturbance in the left plot is identical to the behaviour of the bias in the right plot. In general the bias is small and constant nearly constant in both the horizontal and vertical direction. However, over the horizontal axis the disturbance decreases with an increasing quantity of training data. An identical pattern in observed over the vertical axis, where the magnitude of the bias increases with an increasing target complexity of the input space. From these observations the following conclusions can be drawn:

- Due the usage of only two features, over the horizontal axis the real prediction accuracy does not increase, since the bias is in general small and shows no clearly visible gradient in the horizontal nor the vertical direction). The cause is that the Support Vector Machine has no possibilities of transforming the features into higher dimensional spaces. Consequently, an increasing quantity of training data does not increase the possibility of finding a more optimal decision boundary in a higher dimensional space.

- Visualised by the bias, the pattern of the disturbance of the estimated prediction accuracy is caused by the usage of random seeds in the sample and splitting procedures. In the horizontal direction the disturbance decreases with an increasing quantity of the training data. This is caused due to an increasing statistical equality of the training data sample with the input space, which decreases the relative contribution to the bias by the usage of the random seed in the splitting procedures. Furthermore, the gradient of the decrease in the disturbance along the horizontal axis increases with a decreasing target complexity level of the input space, as the probability that a lesser quantity of training data shows statistical equality with the input space increases. In the vertical direction the disturbance decreases with a decreasing target complexity level of the input space. This is caused due to a decreasing dependency of statistical equality between the training data sample and the input space, of which the reason is explained in the following conclusion.

- As the target complexity level of the input space decreases, the variation of optimal hyper-parameter combinations decreases (see the differences between the impressions of the repeated training sessions in section B.1 and B.2 of appendix B). Due to a decreasing overlap, the final decision boundary becomes less dependent (and finally even completely independent) of the total slack and thus the importance of the training data to reflect the correct overlap decreases. Hence, the consequences of missing an important support vector reduces, which decreases the influence of the usage of the random seed in both the sampling produce and the splitting procedures. This results in a smaller sensitivity of the final hypothesis to the spatial distribution of the training data sample and therefore the variation of optimal hyper-parameters reduces.

# 8
# INSIGHTS INTO SPATIAL VARIATION OF SOIL STRATIFICATION

This chapter present a roadmap, which elaborates the derivation of spatial insights into soil stratification by means of implementing the developed Machine Learning application and subsequently the usage of these insights in order to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements. The roadmap is subdivided into two section: a priori and a posteriori soil stratification insights. Under a priori soil stratification insights the steps that are conducted to derive insights by solely analysing spatial surface settlement measurements are categorised. The sequential steps that follow in order to update the a priori soil stratification insights through implementation of the Machine Learning application are categorised under a posteriori insights. Lastly, although beyond the scope of this thesis, the benefits of implementing the obtained insights in the assessment of dikes and dams is presented. To increase the comprehensibility of this conceptual roadmap, throughout this chapter the example of deriving spatial insights into soil stratification of relevance for primary failure mechanism piping is used.

## 8.1. A Priori Soil Stratification Insights

### 8.1.1. Step 1 - Defining the Target

The first step in the derivation of soil stratification insights is the definition of the target. The target describes the specific spatial information that is required to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements that are associated to piping. Explained in subsection 3.2.2, piping is the phenomenon of water flowing through a self-eroded channel under a water retaining structure. To occur, a permeable sand layer has to cross the full width of the dike or dam. Hence, the target is to determine whether or not a permeable sand layer crosses the full dike or dam width.

### 8.1.2. Step 2 - Defining the Labels

The second step in the derivation of soil stratification insights is the definition of the labels. The characteristics of the labels have to be chosen with respect to the set target. In case of piping, this means that the labels should at least provide a distinction between a soil stratification profile with and without a sand layer. In this thesis the WBI-SOS subsoil scenarios, which are derived for each dike segment part of primary flood defences in the Netherlands, are used as labels. As explained in subsection 3.1.2, each subsoil scenario is a rough stochastic representation of a group of soil stratification profiles within a certain WBI-SOS segment[1], which is derived from an ensemble of local subsoil information (cone penetration tests and boreholes) and global subsoil information from many databases, supplemented by expert knowledge. Subsequently, the subsoil scenarios have to be specified to the targeted dike segment and the primary failure mechanism. This means that subsoil scenarios that are to be expected in the dike segment are updated with local subsoil information. Kruse and Hijma [29] present a detailed elaboration on specifying the subsoil scenarios, called "local schematisation". Figure 8.1 shows the WBI-SOS subsoil scenarios that present the labels that are used in thesis. Note, as explained in subsection 2.3.3, these subsoil scenarios are assumed to be the locally calibrated

[1] A WBI-SOS segment (explained in subsection 3.1.2) differs from a dike segment (explained in 3.2.3).

subsoil scenarios specified to the primary failure mechanism piping. In subsection 9.1.1 the consequences of this assumption are further elaborated.

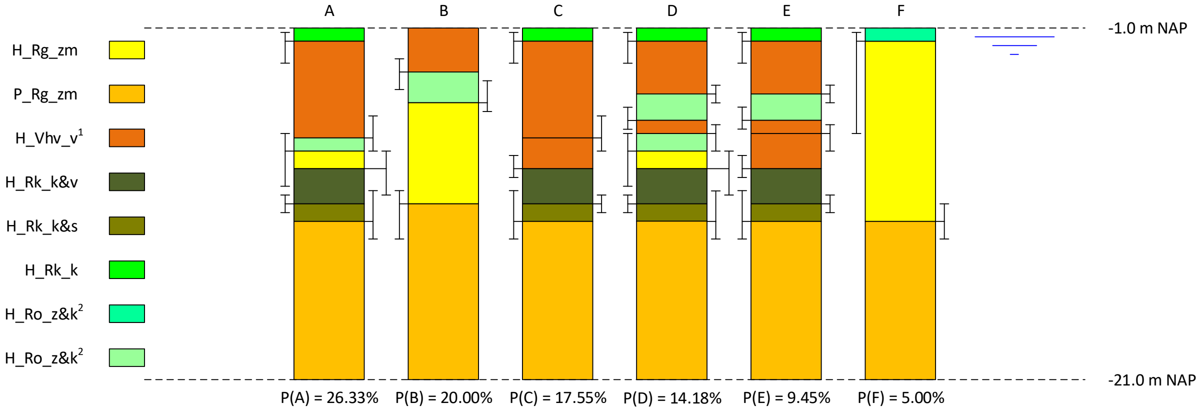


Figure 8.1: Calibrated WBI-SOS subsoil Scenarios to target the primary failure mechanism piping.

### 8.1.3. Step 3 - Analysing the Features

The third step in the derivation of soil stratification insights is the feature analysis. In subsection 6.2.1 it is explained that by analysing the impact of the load on the subsoil scenarios, some of them can directly be distinguished from surface settlement measurements. In this research it was concluded that the behaviour of subsoil scenario B and F differed significantly compared the others. Subsequently, by means of the developed feature extraction method (see subsection 6.2.3), spatial surface settlement measurements combined with information about the spatial load distribution (the self-weight of the dike or dam) can be transformed into features (the input space). With the results of the load impact analysis, parts of the total spatial distribution of the features can be assigned with labels, representing subsoil scenario B and F. Now, the uncertainties regarding spatial soil stratification can be reduced to the unknown spatial distributions of the four remaining WBI-SOS subsoil scenarios shown in Figure 8.2. Since the spatial distributions of the subsoil scenario B and F are known, the probabilities of occurrence of the other subsoil scenarios can be updated to form together again 100% of the spatial uncertainties of soil stratification within the remaining unclassified spatial distribution of the features.

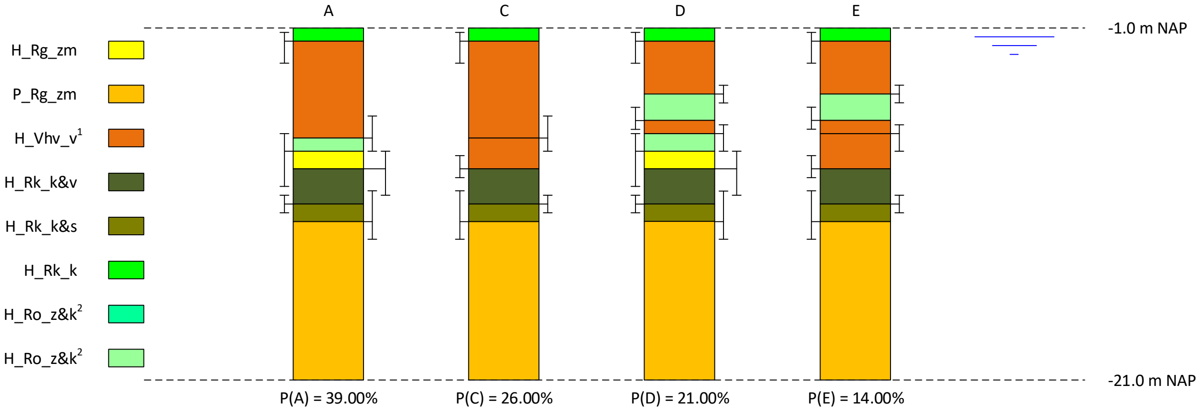


Figure 8.2: Remaining WBI-SOS Subsoil Scenarios that describe the spatial uncertainties of soil stratification by means of their unknown spatial distributions.

Figure 8.3 conceptually illustrates the spatial overview of a dike segment where the a priori soil stratification insights, derived from the feature analysis, are presented. Firstly, the local soil stratification information is assigned with the associated labels. Secondly, after the feature analysis the total area is subdivided into a classified and an unclassified part. The classified part captures the spatial distribution of the excluded subsoil scenarios B and F. The remaining unclassified area visualises the unknown distribution of the subsoil scenarios A, C, D and E.

[2] Figure 8.3 is merely a conceptual example of a priori soil stratification insights, not a real result of underlying calculations.

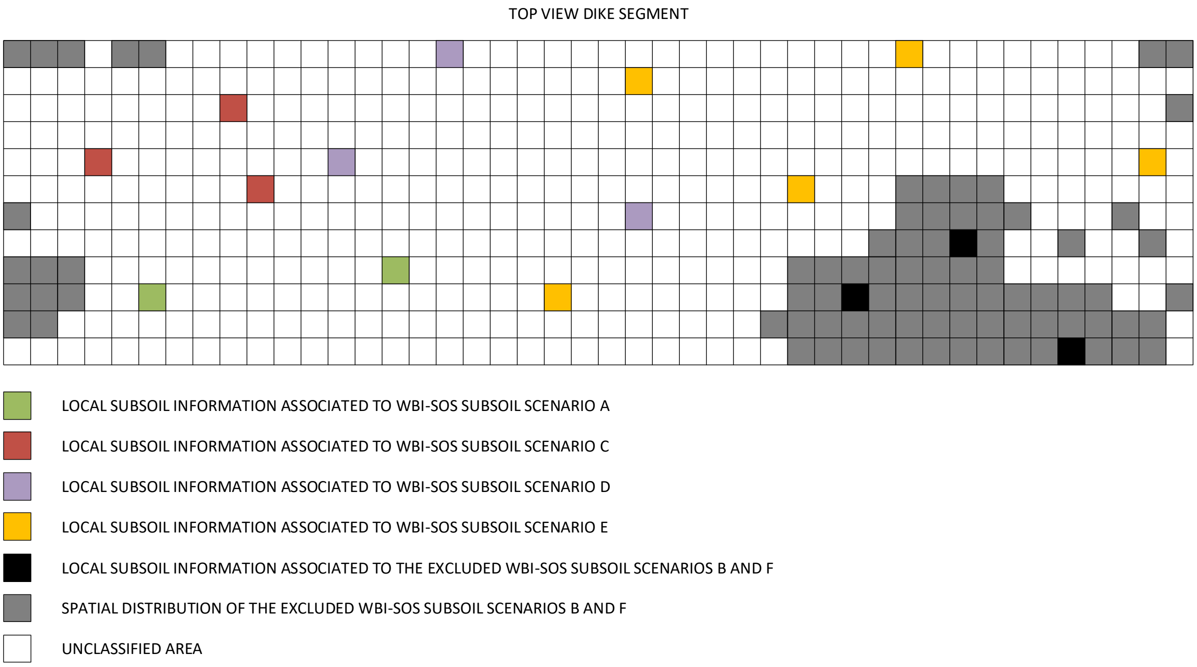

TOP VIEW DIKE SEGMENT
LOCAL SUBSOIL INFORMATION ASSOCIATED TO WBI-SOS SUBSOIL SCENARIO A
LOCAL SUBSOIL INFORMATION ASSOCIATED TO WBI-SOS SUBSOIL SCENARIO C
LOCAL SUBSOIL INFORMATION ASSOCIATED TO WBI-SOS SUBSOIL SCENARIO D
LOCAL SUBSOIL INFORMATION ASSOCIATED TO WBI-SOS SUBSOIL SCENARIO E
LOCAL SUBSOIL INFORMATION ASSOCIATED TO THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
SPATIAL DISTRIBUTION OF THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
UNCLASSIFIED AREA

Figure 8.3: A prior spatial insights in soil stratification obtained by mapping distinguishable and indistinguishable areas in the spatial overview of a dike segment after conducting a load impact analysis.[2]

## 8.2. A Posteriori Soil Stratification Insights

### 8.2.1. Step 4 - Applying Machine Learning

The fourth step in the derivation of soil stratification insights is the application of Machine Learning. After the spatial overview of the dike section is subdivided into a classified and an unclassified area, the unclassified area captures the unknown spatial distribution of the subsoil scenarios A, C, D and E. As explained in subsection 7.3.3, to apply the Machine Learning application the target complexity has to be matched to feature representativity of the input space. To use the Machine Learning application as developed in this thesis, the subsoil scenarios A, C, D and E have to be subdivided into two classes. To comply with the set target, in the composition of the classes a distinction is required between labels that contain an sand layer and ones which do not. Therefore, class 1 and class 2 are formed respectively by subsoil scenario A, D and C, E. Subsequently, the local soil stratification information is classified accordingly as shown in Figure 8.4.

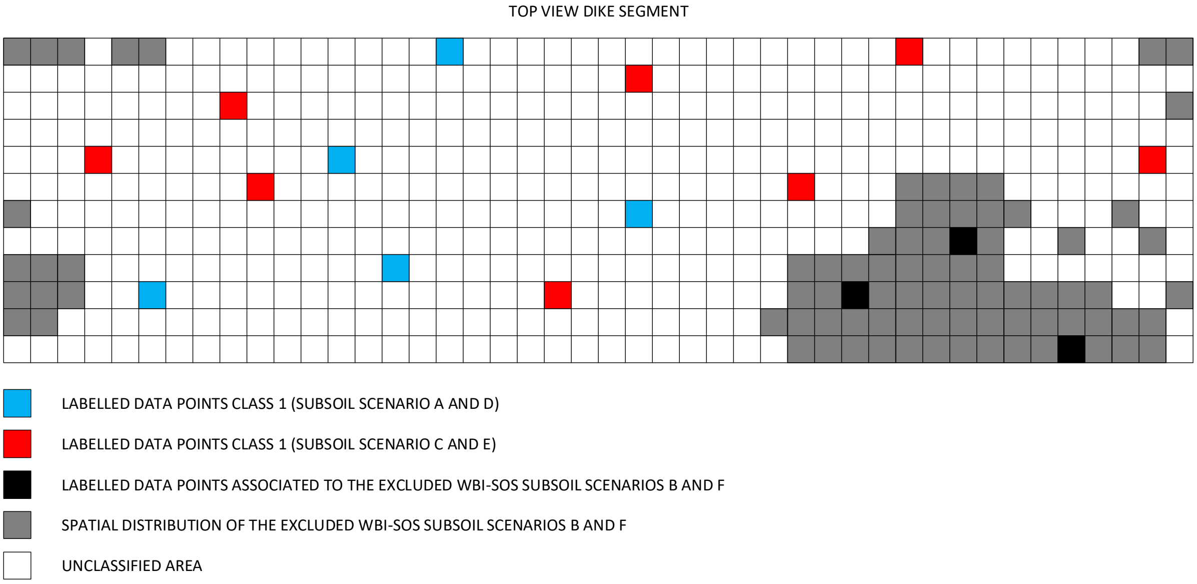

TOP VIEW DIKE SEGMENT
LABELLED DATA POINTS CLASS 1 (SUBSOIL SCENARIO A AND D)
LABELLED DATA POINTS CLASS 1 (SUBSOIL SCENARIO C AND E)
LABELLED DATA POINTS ASSOCIATED TO THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
SPATIAL DISTRIBUTION OF THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
UNCLASSIFIED AREA

Figure 8.4: Classified local soil stratification information in terms of the composed WBI-SOS subsoil scenario classes.

After the local soil stratification is classified, the Machine Learning algorithm is trained. The training data consists of the features and the associated labels as presented by the classified local soil stratification information. With the trained algorithm, the unclassified spatial distribution of the features can be classified. Figure 8.5 conceptually illustrates the spatial overview of a dike segment where the a posteriori soil stratification insights, derived by the results of the Machine Learning application, are presented.

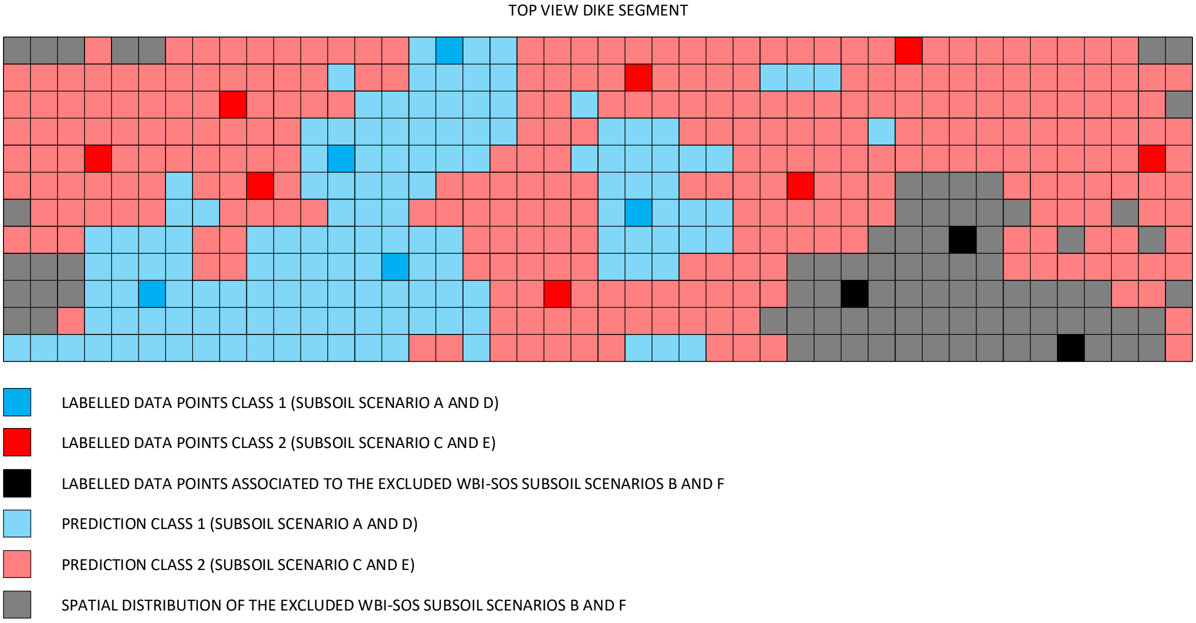

TOP VIEW DIKE SEGMENT
LABELLED DATA POINTS CLASS 1 (SUBSOIL SCENARIO A AND D)
LABELLED DATA POINTS CLASS 2 (SUBSOIL SCENARIO C AND E)
LABELLED DATA POINTS ASSOCIATED TO THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
PREDICTION CLASS 1 (SUBSOIL SCENARIO A AND D)
PREDICTION CLASS 2 (SUBSOIL SCENARIO C AND E)
SPATIAL DISTRIBUTION OF THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F

Figure 8.5: A posteriori insights in spatial soil stratification, obtained from the fully classified spatial overview of a dike segment after applying the Machine Learning application.[3]

### 8.2.2. Step 5 - Reducing Uncertainties Regarding Spatial Variability in Cross-Sectional Reliability Requirements

The fifth step is the reduction of uncertainties regarding spatial variability in cross-sectional reliability requirements. The uncertainties can be reduced through the analysis of the a posteriori soil stratification insights. This is done in two steps. Firstly, dike sections are assigned based on the spatial distribution of the classified areas. Secondly, for each dike segment the probability of occurrence of each possible subsoil scenario is estimated. In Figure 8.6 these steps are conducted on the a posteriori soil stratification insights, as presented in the spatial overview of the dike segment shown in Figure 8.5.

[3] Figure 8.5 is merely a conceptual example of a priori soil stratification insights, not a real result of underlying calculations.

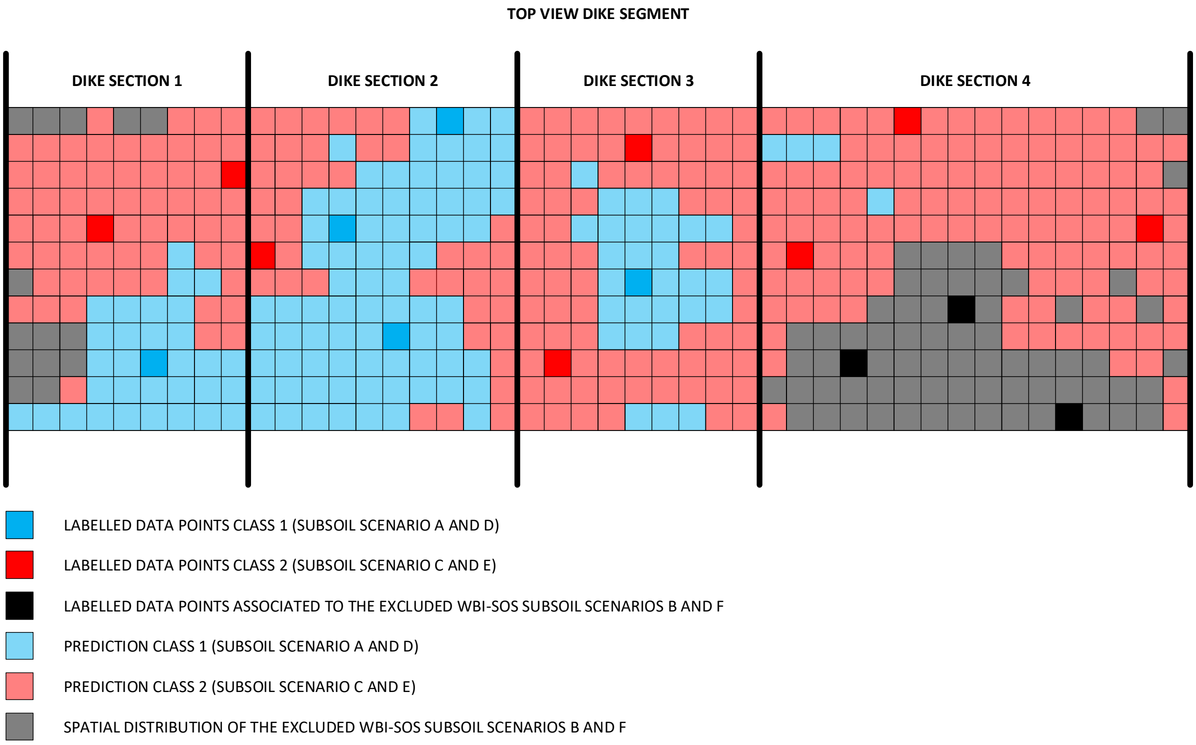

TOP VIEW DIKE SEGMENT
DIKE SECTION 1
DIKE SECTION 2
DIKE SECTION 3
DIKE SECTION 4
LABELLED DATA POINTS CLASS 1 (SUBSOIL SCENARIO A AND D)
LABELLED DATA POINTS CLASS 2 (SUBSOIL SCENARIO C AND E)
LABELLED DATA POINTS ASSOCIATED TO THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F
PREDICTION CLASS 1 (SUBSOIL SCENARIO A AND D)
PREDICTION CLASS 2 (SUBSOIL SCENARIO C AND E)
SPATIAL DISTRIBUTION OF THE EXCLUDED WBI-SOS SUBSOIL SCENARIOS B AND F

Figure 8.6: Example of assigning dike sections and updating the probability of occurrence of each WBI-SOS subsoil scenario, based on the spatial distributions of the classified areas.[4]

Subsequently, by analysing the spatial distribution of the subsoil scenarios, it can be seen that only dike section 2 is prone to piping, since a continuous sand layer crosses the full width of the dike. Whilst in the other dike segments sand layers are present, they don't form corridors that cross the full width of the dike. Consequently, piping cannot occur in those segments.

After conducting the analysis of the a posteriori soil stratification insights, the uncertainties regarding spatial variation have been reduced due to the visualisation of the spatial distribution of the subsoil scenarios. This effects the cross-sectional reliability requirements through the length-effect. As explained in subsection 3.3, the magnitude of the default values of $a$ (which presents the fraction of the dike segment that is sensitive to the targeted primary failure mechanism) are often based on expert judgment given the local circumstances per dike segment. Given the results of the analysis of the a posteriori soil stratification insights, the estimated $a$ value can be updated or strengthened. This reduces soil stratification uncertainties that are currently incorporated in the magnitudes of the length-effect factors and thereby improves the accuracy the translation of the safety standard into cross-sectional requirements.

### 8.2.3. Step 6 - Implementing A Posteriori Insights in the Assessment of Dikes and Dams

The potential application of the a posteriori soil stratification information is broader than only reducing the uncertainties regarding spatial variability in cross-sectional reliability requirements of dike and dams. Therefore, although beyond the scope of this thesis, an example is given regarding the implementation in the assessment of dike and dams.

During the assessment of dike of dams, per dike section the resistance against piping is mapped by calculating the resistance of the normative cross-section for multiple subsoil schematisations. Each subsoil schematisation is based on a locally calibrated WBI-SOS subsoil scenarios that are present within the dike segment. Due to the lack of knowledge on the spatial distribution of the probability of occurrence per subsoil scenario, a major downside of this method is the validity range of each subsoil schematisation is set to the total length of the dike section. Consequently, if the safety of the dike section appears to be insufficient, the full dike section needs to be reinforced.

[4] Figure 8.6 is merely a conceptual example of a priori soil stratification insights, not a real result of underlying calculations.

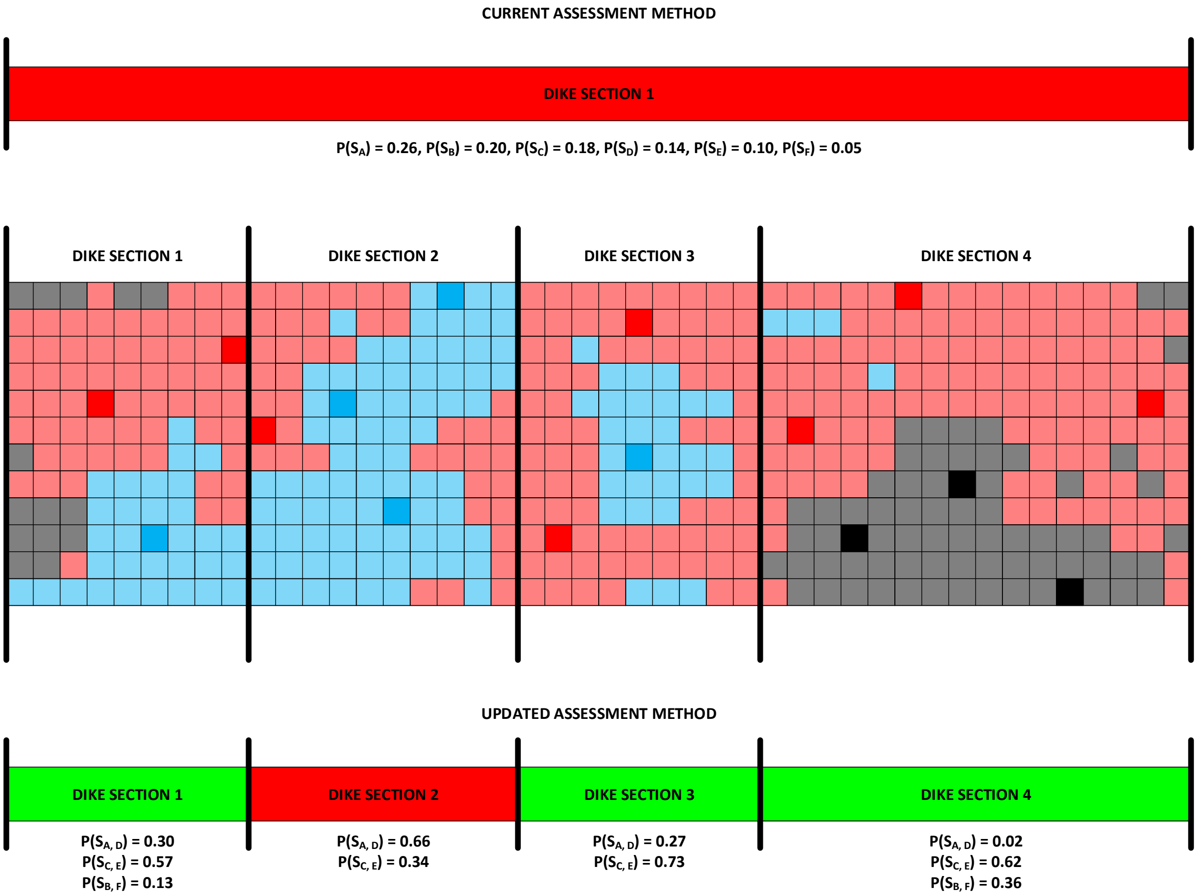


Figure 8.7: Updating the assessment accuracy through the analysis of the a posteriori soil stratification information.

In Figure 8.7 the currently used method is illustrated in the top plot. In the dike section the WBI-SOS subsoil scenarios with their corresponding probabilities of occurrence are presented. Based on each a subsoil schematisation is defined. After hypothetical calculations are performed to determine the resistance of the normative cross-section against piping, it turned out that the average safety of the dike section was not sufficient (indicated by red).

By analysing the a posteriori soil stratification insights the dike section layout can be updated with respect to the spatial distributions of the subsoil scenarios, as show in the middle plot of Figure 8.7. The results of this analysis are implemented in the assessment method and are shown in the bottom plot of Figure 8.7. Per dike section in the updated layout the probabilities of occurrence of the subsoil scenarios are calculated. Through the analysis of the spatial distribution of the subsoil scenarios, it can be seen that only dike section 2 is prone to piping. Given the results of the assessment calculation, dike section 2 is insufficiently safe for piping and thus should be reinforced. In the other dike sections piping cannot occur, hence, the safety in those dike sections is at the required level (indicated by green).

In conclusion, by means of the a posteriori subsoil information it becomes possible to update the dike section layout and to analyse which of the updated dike sections are prone to the targeted primary failure mechanism. As this information can be used to for a better estimation of the subsoil scenario probabilities, it improves the accuracy of the assessment and effectiveness of potential future reinforcement measures.

# IV

# CLOSURE & CONCLUSIONS

# 9 DISCUSSION

In this chapter the methodology and the derived results presented in chapter 6 and 7 are discussed in respectively section 9.1 and 9.2.

## 9.1. Evaluation of the Methodology

The topics addressed in this section are ranked with respect to structure of the report.

### 9.1.1. Quality of the WBI-SOS Subsoil Scenarios

Explicitly mentioned by the WBI-SOS programme is that the WBI-SOS subsoil scenarios are not locally schematised. Also, they are not specified towards the primary failure mechanism that is targeted by the Machine Learning application. Therefore, the variation per class in the feature space is most likely overestimated. This implies that the assumed target complexity level of the input space is too high, which means that in this thesis a more difficult learning problem is sketch compared to the learning problem in reality.

### 9.1.2. Quality of the General Oedometer Test Data Set

The data set of oedometer tests could not be used to indication spatial variety of soil layer parameters. Therefore, no soil parameter value distributions are included in the D-Settlement calculations, resulting in an underestimation of the variation per class in the feature space. In reality spatial variation in soil parameters in encountered. This implies that the assumed target complexity level of the input space is too low, which means that in this thesis a more simplified learning problem is sketch compared to the learning problem in reality.

### 9.1.3. Quality of D-Settlement

In the usage of D-Settlement it was not possible to include an impermeable boundary at the top of soil column to simulates the dike body. The result of this limit was that the excess pore water pressure could flow away in two directions, accelerating the consolidation process. In this research it affected all imported soil columns, therefore it did not have any negative effects on the accuracy of the feature extraction method (all features are equally transformed in the feature space) and thus the results remained unbiased.

### 9.1.4. Quality of the Consolidation Stage Transit Detection

If the time series of local measurements are too short, the transition point between the primary and secondary consolidation phase has not been reached and cannot be detected. In assumption that the consolidation rates of the input space are equal, the developed feature extraction method is capable of extraction features from settlement curves. In case of unequal consolidation rates in combination with estimated measurement errors, *Fit Constant a* and *Fit Constant b* can become biased. Subsequently, this bias affects negatively the accuracy of the feature extraction method, which again negatively affects the prediction accuracy of the algorithm.

## 9.2. Evaluation of the Results

In this section the effects on the results by certain decisions and assumption are discussed. The topics addressed in this section are ranked with respect to structure of the report.

### 9.2.1. Quality of the 1D Approximation of the Subsoil

In this thesis it is assumed that the subsoil can be subdivided into individual soil columns, which do not interact with the adjacent soil columns. Due to shear forces, in reality soil columns do interact with each other. The feature extraction method does not include any component that takes these shear forces into account. However, in applications to real cases the feature extraction method has to be calibrated to the measured settlement curves that are related to the local subsoil measurements. In case of a well-performed calibration, the shear force components are indirectly included into the model, since their contribution of is already reflected in the measured settlement curves. The effects of the spatial variation in the shear forces is difficult to include in a locally calibrated model. Hence, this component will cause an inevitable error that has to be accepted.

The consequence of the 1D approximation in combination with a classification problem, are hard borders between the predicted classes of WBI-SOS subsoil scenarios. Due to the nature of natural processes, in reality these hard borders are not to be expected. In any estimation of the fraction of the dike segment that is sensitive to the targeted primary failure mechanism or update of the dike section layout based on the analysis of the spatial distribution of the predicted classes, this should be taken into account. To improve this effect, one can decide to express the prediction per grid cell in probabilities of each subsoil scenario class. Based on such outcome, a statistical soft border between the distributions of the classes is created.

### 9.2.2. Quality of Fitting the Settlement Curves

It can be concluded that the fit by the Power Model is accurate for all the WBI-SOS subsoil scenarios, expect for subsoil scenario F. Since subsoil scenario F is composed of only 0.5 m of compressible soil, being subjected to load combinations resulting in extreme small settlements. In combination with numerical instability of D-Settlement, it was not possible to indicate the transition between the primary and secondary consolidation stage. However, due the possibility of distinguishing subsoil scenario F from spatial surface settlement measurements prior to any analysis by a Machine Learning application, in this research it did not form an obstacle. One should be careful with these results nonetheless. Basically, in this research the integrated a,b,c-isotache model of D-Settlement is being fit. It is crucial to calibrate such calculations first to local measured data before making a conclusion about the fit quality.

Since measurement uncertainties were excluded in this research, stochastic noise was excluded from the calculated. Therefore, it is most likely that the prediction accuracy is slightly overestimated. As explained in subsection 5.3.5, there are tuning techniques and algorithms that are highly effective in dealing with stochastic noise. Hence, the impact of stochastic noise on the prediction accuracy is difficult to assess prior to testing.

### 9.2.3. Quality of Extracting the Features

In subsection 6.3 the prediction accuracy of the feature extraction method is approximated to be 92%. Since no statistic noise was included, as explained in subsection 9.2.2, this means that the method itself causes the error of 8%. The composition of the four formulas is found by iterating different function types (linear, power, logarithmic, etc.) to the range of the corresponding fit constants (*Fit Constant a* and *Fit Constant b*) per WBI-SOS subsoil scenario. The final function that was chosen per term in the formulas is based on its ability to fit *Fit Constant a* and *Fit Constant b* under the constraints that are mentioned in subsection 6.2.3. The parameters found after the fit to load scenario 1 (equation 6.2 and 6.4) were used as fixed input parameters (without the flexibility of their corresponding fit error) for the fit to load scenario 2 (equation 6.3 and 6.5). The result is that the fit to Load Scenario 1 determines the value of $x_{feature\ 1}$ and $x_{feature\ 2}$. Consequently, in the fit to load scenario 2, the composition of equation 6.3 and 6.5 was chosen such that they provide the best calibration to the found $x_{feature\ 1}$ and $x_{feature\ 2}$. Consequently, this gives the fit to load scenario 1 significant dominance over the fit values of $x_{feature\ 1}$ and $x_{feature\ 2}$. However, probably the largest share of the training data set will fall in the domain of load scenario 2. Therefore, it is most likely that the uncertainties of the fit to load scenario 2 will determine the deviation of the feature extraction method as a whole. Hence, one can conclude that even though the relative coefficient of variation of the fit values of $x_{feature\ 1}$ and $x_{feature\ 2}$ are of the same order as the measured coefficients of variation, they are less reliable. A more accurate approximation of the accuracy of the feature extraction method is reflected by the found coefficients of variation, resulting in a approximated accuracy of 92%.

### 9.2.4. Quality of the Measured Prediction Accuracy

During the learning process and the testing of the trained algorithms, the prediction accuracy is measured by the error measures: $F_1$ Score and the Matthews Correlation Coefficient. The Matthews Correlation Coefficient is considered to be the best error measure for binary classification problems, since it measures the prediction quality of both labels, undisturbed by any skewness of their distribution. In contrast, the $F_1$ Score used during the grid search cross-validation does only measure the quality of only one label. This means that the score is only representative if both labels are equally distributed among the training data.

In this research the ratio between class 1 and 2 for the training data samples is around 60% to 40%. This indicates that the performance measure is slightly skewed towards class 2. Consequently, the $F_1$ Score, used to determine the best hypothesis during the grid search cross-validation, has favoured hypothesis that provided better prediction of class 2.

# 10 CONCLUSIONS AND RECOMMENDATIONS

Firstly, in this chapter the conclusions of this research are summarised in section 10.1. Secondly, in section 10.2 it is discussed if the presented proof of concept has achieved the objective and how the results of this thesis might affect the current Dutch flood control philosophy. Lastly, in section 10.3 the recommendations regarding improvement of the methodology itself and the application to real cases are given.

## 10.1. Conclusions

In this section the conclusions of the research are summarised. Given the answers to the research questions it will basically become clear whether or not the objective of this thesis has been achieved.

*How can a combination of surface settlement information and associated load information be transformed into Machine Learning features?*

This thesis presents a feature extraction method that is capable of extracting features that characterise the compressed subsoil. By analysing the impact of the load on settlement curves, four generic formulas have been fit. Subsequently, given any settlement curve and the corresponding information about the magnitude of the new load and the preload, with these formulas features are extracted with an accuracy of approximately 92%. The formulas are applicable to the full range of settlement curves and corresponding load combinations that are to be expected to occur within the fictitious heightening case.

*How is the prediction accuracy of the Machine Learning application influenced by the quantity of the training data and the target complexity of the input space?*

Firstly, the target complexity of the input space had to be matched with the quantity of the training data. This means that the features have to be capable of reflecting the target complexity of the input space. It was observed that the two generated features from the input space were not capable of representing the complexity of the input space in terms of individual WBI-SOS subsoil scenarios. Therefore, the target complexity of the input space was reduced from individual subsoil scenarios into two distinguishable classes of subsoil scenarios, which appeared to be a proper match to the available quantity of training data.

Secondly, the sensitivity of the prediction accuracy with respect to variation in the target complexity level of the input and quantity of the training data was determined. By means of sample bounds, constrained layer distributions were used to create classified environments of varying target complexity levels. By increasing the bounds in steps of 5% from the 45th - 55th to the 0th - 100th percentiles, gradually the overlap between the classes was increased. To test the influence of the training data quantity on the prediction accuracy, for each classified environment corresponding to a certain target complexity level training data has been sampled, of which the size was gradually increased in steps of 50 from 50 to 500 training data points.

The results of the calculations showed prediction accuracies ranging between 100% (45$^{\text{th}}$ - 55$^{\text{th}}$ percentiles) to 75% (0$^{\text{th}}$ - 100$^{\text{th}}$). The real prediction accuracy did not increase with larger quantities of training data by using only two features but increased with a decreasing complexity of the training data. In contrast, the measurable estimated prediction accuracy showed dependency to the quantity of training data and the target complexity level of the input space. This dependency was presented by the bias between the estimated and the real prediction accuracy and is caused by the usage of random seeds in the sample and splitting procedures. The bias decreases significantly with an increasing quantity of training data, because the relative contribution of the usage of the random seed in the splitting procedures reduces with an increasing statistical equality between the training data and the input space. The gradient of the decrease is positively influenced by a decreasing target complexity level of the input space, as the probability that lesser quantity of training data shows statistical equality with the input space increases. Furthermore, a decreasing target complexity level of the input space resulted in a decreasing bias due to a decreasing dependency of statistical equality between the training data and the input space and also in a decreasing variation of optimal hyper-parameter combinations.

*How are insights into spatial variation of soil stratification derived and how can they be used to improve the approach of deriving technical requirements of dikes and dams?*

With respect to applying Machine Learning, insights into spatial variation of soil stratification can be distinguished in a priori and a posteriori soil stratification insights. A priori insights can be derived through the analysis of extracted features from spatial surface settlement measurements. The results are directly distinguishable subsoil scenarios from spatial surface settlement measurements, which can be located in a spatial overview of a dike segment.

The a posteriori insights are obtained by applying Machine Learning. A Machine Learning algorithm is capable of analysing spatial surface settlement measurements and to predict the spatial distribution of the remaining subsoil scenarios in a spatial grid, which presents the a posteriori soil stratification insights. By analysing the spatial distribution of the subsoil scenarios, corridors can be assigned which are prone to the targeted primary failure mechanism. Given this information, the estimated fraction of the dike segment sensitive to the targeted primary failure mechanism ($a$) can be updated or strengthened, which reduces the uncertainties that are incorporated in the current approximation of the length-effect factors.

## 10.2. Proof of Concept

In this section it is discussed if the presented proof of concept has achieved the objective and how the results of this thesis might affect the current Dutch flood control philosophy. The objective of this thesis reads as follows:

*Providing the proof of concept that a Machine Learning application, by learning locally measured information and analysing high spatial resolution surface settlement data, can increase insights into spatial variation of soil stratification below dikes and dams founded on heterogeneous subsoils, in order to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements.*

In this thesis the methodology used to derive insights in spatial variation of soil stratification relevant for the primary failure mechanism piping can be summarised in four steps:

1. The first step is the definition of the target, which describes the specific spatial information that is required to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements that are associated to piping. In case of piping, the target is to determine whether or not a permeable sand layer crosses the full dike or dam width.

2. The second step is the definition of the labels, which characteristics have to be chosen with respect to the set target. In case of piping, this means that the labels should at least provide a distinction between a soil stratification profile with and without a sand layer. In this thesis WBI-SOS subsoil scenarios are used as labels, which are stochastic representations of groups of similar soil stratification profiles.

3. The third step is the feature analysis. By means of the developed feature extraction method, spatial surface settlement measurements combined with information about the spatial load distribution are transformed into features. By analysing the impact of the load on the subsoil scenarios, certain subsoil scenarios can directly be distinguished from the spatial distribution of features. Subsequently, the spatial overview of a dike segment can be subdivided into a classified and an unclassified part.

4. The fourth step is the application of Machine Learning. To apply Machine Learning, the target complexity has to be matched to feature representativity of the input space. Therefore, the remaining non-directly distinguishable subsoil scenarios have to be subdivided into two classes. To comply with the set target, in the composition of the classes a distinction is required between labels that contain a sand layer and ones which do not. Subsequently, the local soil stratification is classified accordingly. Based on the training data, which consists of the features and the associated labels as presented by the classified local soil stratification information, the Machine Learning algorithm is trained. After the training, the unclassified spatial distribution of the features can be classified. The result is a fully classified spatial overview of the dike segment, which visualises the spatial distribution of the subsoil scenarios.

Given the derived insights in spatial variation of soil stratification, uncertainties regarding spatial variability within the length-effect factors can be reduced. Hence, it can be stated that insights into spatial variation of soil stratification derived by the Machine Learning application can be used to reduce uncertainties regarding spatial variability in cross-sectional reliability requirements and therefore the objected has been achieved.

The results of this methodology are not only applicable in reducing uncertainties regarding spatial variability in cross-sectional reliability requirements. Another potential application is for example to improve the assessment method for dikes and dams. By updating the layout of the dike sections and analysing which dike section is prone to the targeted primary failure mechanism, better estimations of subsoil scenarios probabilities can be made. This improves the accuracy of the assessment method and effectiveness of potential future reinforcement measures. The Machine Learning application as presented in this thesis shows that there is potential to strengthen the currently used safety philosophy in the Netherlands by integrating and analysing multiple data sources. To properly exploit this potential, structured data mining and standardised data storage within the Dutch flood control system is required. The WBI-SOS subsoil scenarios is a good start, however a comprehensive infrastructure that systematically collects and categorises all available information is yet non-existent. The creation of such an infrastructure is crucial, not only to enhance the possibilities to upgrade the current flood control system, but also to be prepared for the problems of tomorrow.

## 10.3. Recommendations

In this section the recommendations regarding improvement of the methodology itself and the application to real cases are presented in respectively subsection 10.3.1 and 10.3.2.

### 10.3.1. Adjustments of the Methodology

To comply with the structure of the report, recommendations regarding the improvement of the methodology are subdivided into the categories: *Feature Extraction Method* and *Learning Model*. Within each category, the recommendations are ranked from highly advised to adjust to merely suggested order to improve the methodology.

#### Feature Extraction Method

- Improve the computation of fictitious surface settlement measurements to decrease numerical uncertainties and to offer more freedom in simulating realistic circumstances with respect to excess pore water pressure flow.

- Increase the number of extracted features to achieve better performance while dealing with a high target complexity of the input space. A first suggestion is to extract features from the secondary consolidation phases.

- Improve the accuracy of the feature extraction method by fitting the four formulas at the same time over both load scenarios.

- Study the influence on the performance by using a non-physically related feature extraction method. For example, it would be interesting to investigate features that are extracted by algorithms used for image recognition. Such methods are capable of detecting promising features from metadata that is hidden from direct observations of settlement curves.

**Learning Model**

- Diminish the bias of the estimated prediction accuracy of the final hypothesis, caused by using a random seed during the sampling and splitting procedures. Through verification, one should measure if the training data sample and any duplications (training, validation and test sets) are stochastically equal to the WBI-SOS subsoil scenarios. Training based on sets that show statistically inequality should not be allowed.
- Change the $F_1$ Score by the Matthews Correlation Coefficient to measure the performance of the hypothesis during the grid search cross-validation.
- Increase the hypothesis set by adding different learning algorithms.
- Study the influence on the prediction accuracy of varying grid resolutions of the coarse and fine grid used in the grid search cross-validation.
- Study the influence of varying split ratios on the prediction accuracy.
- Study the influence on the prediction accuracy of standardising versus normalising the training data.
- Adjust the target function, $f$, by a target distribution $P(y \mid x)$ and a corresponding error measure, to obtain an output space in terms of a probabilities of class 1 or class 2.

### 10.3.2. Adjustments to the Methodology for Practical Usage

The recommendations regarding the application of the methodology all applicable to the category *Feature Extraction Method*. Within this category, the recommendations are ranked from highly advised to adjust to merely suggested order to improve the methodology.

**Feature Extraction Method**

- In the analysis of the impact of the load, calibrate the response of the fictitious subsoil, represented by synthetic surface settlement measurements, to real measured surface settlement measured in the dike segment, in order to obtain a reliable feature extraction method and an environment that represents the subsoil. Adding settlement plate information would increase the understanding of extracting the response of the subsoil itself from the real measured surface settlement measurements.
- Instead of using the global WBI-SOS subsoil scenarios use the locally schematised WBI-SOS subsoil scenarios, which are specified with respect to local subsoil information and the primary failure mechanism that is being targeted by the Machine Learning application.
- Include spatial variety of soil layer parameters and expected measurements uncertainties into the calculations of the synthetic surface settlement curves.
- Develop a method capable of indicating the consolidation rate of each measured settlement curve. This takes away the minimum time span required for Batched Learning. Instead, one could apply an Online Learning approach, which updates the quality of its own prediction over time.

# BIBLIOGRAPHY

[1] LmFit. `https://lmfit.github.io/lmfit-py/`. Accessed: 2018-08-02.

[2] scikit-learn. `https://scikit-learn.org/stable/`. Accessed: 2018-09-30.

[3] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from Data*. 2012.

[4] ASTM. Standard Practice for Thin-Walled Tube Sampling of Soils for Geotechnical, 2015.

[5] Mariette Awag and Rahul Khanna. *Efficient Learning Machines - Theories, Concepts, and Applications for Engineerings and System Designers*. ApressOpen, 2015.

[6] Laing Barden. Primary and secondary consolidation. *Géotechnique*, 18(1):1–22, 1968. ISSN 01651250. doi: 10.1016/B978-0-444-98822-5.50012-0.

[7] Claudia Beleites, Ute Neugebauer, Thomas Bocklitz, Christoph Krafft, and Jürgen Popp. Sample Size Planning for Classification Models. *Analytica Chimica Acta*, 760(June 2012):25–33, 2013. ISSN 00032670. doi: 10.1016/j.aca.2012.11.007.

[8] Shai Ben-David and Shai Shalev-Shwartz. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 9781107057135. doi: 10.1017/CBO9781107298019.

[9] Asa Ben-Hur and Jason Weston. A User's Guide to Support Vector Machines. In *Methods in Molecular Biology*, chapter 13, pages 223–239. Humana Press, 2010.

[10] Stephen Billets. Environmental Technology Verification Report Soil Sampling Technology Simulprobe Technologies , Inc . Core Barrel Sampler. Technical Report August, 1998.

[11] E Boser, N Vapnik, and Isabelle M Guyon. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the fifth annual workshop on Computational learning theory – COLT '92*, pages 144–152, 1992. ISSN 0-89791-497-X. doi: 10.1145/130385.130401.

[12] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal Classifier for Imbalanced Data Using Matthews Correlation Coefficient Metric. *PLoS ONE*, 12(6):1–17, 2017. ISSN 19326203. doi: 10.1371/journal.pone.0177678.

[13] Christopher J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognitions. *Data Mining and Knowledge Discovery 2*, pages 121–167, 1998. ISSN 16000838. doi: 10.1111/sms.12977.

[14] Davide Chicco. Ten Quick Tips for Machine Learning in Computational Biology. *BioData Mining*, 10(1): 1–17, 2017. ISSN 17560381. doi: 10.1186/s13040-017-0155-3.

[15] Carinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 08856125. doi: 10.1007/BF00994018.

[16] Braja M. Das and Khaled Sobhan. *Principles of Geotechnical Engineering*, volume 39. Cengage Learning, 8 edition, 2012. ISBN 9781133108672.

[17] Deltares. D-Settlement - User Manual, 2016. ISSN 1920-7476.

[18] E. J. den Haan. *Vertical Compression of Soils*. PhD thesis, Delft University of Technology, 1994.

[19] Evert J. den Haan. A History of the Development of Isotache Models. 2007.

[20] M. Hijma and K.S. Lam. Globale Stochastische Ondergrondschematisatie (WTI-SOS) voor de Primaire Waterkeringen. Technical report, Deltares, 2015.

[21] Daniel Hillel. *Introduction to Soil Physics*, volume 150. Academic Press, 1982. ISBN 0-12-348520-7.

[22] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. 2003. ISSN 02632764. doi: 10.1177/02632760022050997.

[23] J. Jiang, Q. Chen, and S. Nimbalkar. Field Data Based Method for Predicting Long-Term Settlements. *American Journal of Engineering and Applied Sciences*, 9(3), 2016. ISSN 19417039. doi: 10.3844/ajeassp.2016.466.476.

[24] R.B. Jongejan. WBI2017 Code Calibration. Technical Report June, 2017.

[25] S N Jonkman, R.E. Jorissen, T Schweckendiek, and J.P. van den Bos. *Flood Defences*. Delft University of Technology, Delft, 2018.

[26] Wim Kanning. *The Weakest Link*. PhD thesis, 2009.

[27] M. Kok, R. Jongejan, M. Nieuwjaar, and I. Tanczos. Fundamentals of Flood Protection. Technical report, 2017.

[28] S.B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, pages 249–268, 2007. ISSN 15729540. doi: 10.1007/s10751-016-1232-6.

[29] Gerard Kruse and Marc Hijma. WTI 2017: Handleiding Lokaal Schematiseren met WTI-SOS. Technical report, Deltares, 2015.

[30] P Lubking. Dijkversterking Bergambacht- Ammerstol-Schoonhoven - Uitvoeringsrapport Grondlichamen. Technical report, Deltares, 2009.

[31] Tom Lunne, Peter K. Robertson, and John J.M. Powell. *Cone Penetration Testing in Geotechnical Practice*, volume 46. 1997. ISBN 041923750X. doi: 10.1007/s11204-010-9072-x.

[32] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1 edition, 1997. ISBN 0070428077.

[33] Andrew Y. Ng. Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance. *Twenty-first international conference on Machine learning - ICML '04*, 2004. ISSN 0253-0465. doi: 10.1145/1015330.1015435.

[34] F.Y. Osisanwo, J.E.T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi. Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3):128–138, 2017. ISSN 22312803. doi: 10.14445/22312803/IJCTT-V48P126.

[35] Nicola W. Petty and Shane Dye. Triangular Distributions, 2013.

[36] Sebastian Raschka. *Python Machine Learning*. Packt Publishing Ltd., 2015. ISBN 9781783555130.

[37] Seyed Navid Resalat and Valiallah Saba. A Study of Various Feature Extraction Methods on a Motor Imagery Based Brain Computer Interface system. *Basic and Clinical Neuroscience*, 7(1):13–20, 2016. ISSN 2008126X.

[38] Peter K. Robertson. Soil behaviour type from the CPT: an update. *2nd International Symposium on Cone Penetration Testing*, (May):8 p., 2010. ISSN <null>.

[39] Arthur L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 44(1):206–226, 2000. ISSN 0018-8646. doi: 10.1147/rd.441.0206.

[40] Timo Schweckendiek. *On Reducing Piping Uncertainties: A Bayesian Decision Approach*. PhD thesis, Delft, 2014.

[41] R Slomp. Flood Risk and Water Management in the Netherlands. Technical report, RWS VWL, 2012.

[42] L.M.Th Swart. How the Up-To-Date Height Model of The Netherlands (AHN) Became a Massive Point Data Cloud. *Management of Massive Point Cloud Data: Wet and Dry*, pages 1–18, 2010.

[43] Robert 't Hart. Fenomenologische Beschrijving. Technical report, Deltares, 2018.

[44] Karl Terzaghi. *Theoretical Soil Mechanics*. 1943.

[45] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Elsevier, 2 edition, 2003. ISBN 0126858756.

[46] V. N. Vapnik and A. Ya Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Measures of Complexity: Festschrift for Alexey Chervonenkis*, XVI(2):11–30, 2015. ISSN 0040-585X. doi: 10.1007/978-3-319-21852-6_3.

[47] V. N. Vapnik and A. Ya. Lerner. Pattern Recognition Using Generalized Portraits. *Avtamarika I Telernckl-lanjka*, 24(6):774–780, 1963.

[48] Ronny Vergouwe. Veiligheid Nederland in Kaart. Technical report, Royal Haskoning DHV, 2014.

[49] Arnold Verruijt. *Soil Mechanics*. 2006.

[50] J. K. Vrijling, Timo Schweckendiek, and Wim Kanning. Safety Standards of Flood Defences. *Proceedings of the 3rd International Symposium on Geotechnical Safety and Risk, ISGSR 2011, Munich, Germany*, pages 67–84, 2011.

[51] H.J.T. Weerts, P Cleveringa, J.H.J. Ebbing, F.D. de Lang, and W.E. Westerhoff. De Lithostratigrafische Indeling van Nederland – Formaties uit het Tertiair en Kwartair. Technical report, TNO, 2000.

[52] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou. A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(2W3):281–285, 2014. ISSN 16821750. doi: 10.5194/isprsarchives-XL-2-W3-281-2014.

# A
# FITTED AND CALCULATED RESULTS

## A.1. Fitted *Fit Constant a* and *Fit Constant b*

Table A.1: *Fit Constant a* and *Fit Constant b* of the Power Model, fitted to the settlement curves predictions of the WBI-SOS subsoil scenarios (without layer interface distributions), each subjected to the 30 load combinations.

| | | SSA | | | | SSC | | | |
|---|---|---|---|---|---|---|---|---|---|
| **New load** | **Preload** | $a$ | $\sigma_a$ | $b$ | $\sigma_b$ | $a$ | $\sigma_a$ | $b$ | $\sigma_b$ |
| [kPa] | [kPa] | [-] | [-] | [-] | [-] | [-] | [-] | [-] | [-] |
| 2.375 | 0.000 | $1.79\cdot10^{-3}$ | $7.33\cdot10^{-5}$ | $5.74\cdot10^{-1}$ | $4.35\cdot10^{-3}$ | $1.05\cdot10^{-3}$ | $6.10\cdot10^{-5}$ | $6.22\cdot10^{-1}$ | $5.87\cdot10^{-3}$ |
| 7.125 | 0.000 | $5.29\cdot10^{-3}$ | $1.35\cdot10^{-4}$ | $5.22\cdot10^{-1}$ | $2.81\cdot10^{-3}$ | $3.06\cdot10^{-3}$ | $1.28\cdot10^{-4}$ | $5.70\cdot10^{-1}$ | $4.33\cdot10^{-3}$ |
| 9.500 | 0.000 | $7.06\cdot10^{-3}$ | $1.45\cdot10^{-4}$ | $5.10\cdot10^{-1}$ | $1.90\cdot10^{-3}$ | $3.83\cdot10^{-3}$ | $1.38\cdot10^{-4}$ | $5.66\cdot10^{-1}$ | $3.84\cdot10^{-3}$ |
| 12.000 | 0.000 | $9.37\cdot10^{-3}$ | $1.72\cdot10^{-4}$ | $4.96\cdot10^{-1}$ | $2.08\cdot10^{-3}$ | $4.90\cdot10^{-3}$ | $1.80\cdot10^{-4}$ | $5.56\cdot10^{-1}$ | $3.92\cdot10^{-3}$ |
| 17.000 | 0.000 | $1.40\cdot10^{-2}$ | $1.92\cdot10^{-4}$ | $4.77\cdot10^{-1}$ | $1.60\cdot10^{-3}$ | $6.59\cdot10^{-3}$ | $2.19\cdot10^{-4}$ | $5.49\cdot10^{-1}$ | $3.64\cdot10^{-3}$ |
| 22.000 | 0.000 | $1.91\cdot10^{-2}$ | $2.35\cdot10^{-4}$ | $4.61\cdot10^{-1}$ | $1.45\cdot10^{-3}$ | $8.65\cdot10^{-3}$ | $2.91\cdot10^{-4}$ | $5.38\cdot10^{-1}$ | $3.68\cdot10^{-3}$ |
| 27.000 | 0.000 | $2.40\cdot10^{-2}$ | $3.06\cdot10^{-4}$ | $4.51\cdot10^{-1}$ | $1.54\cdot10^{-3}$ | $1.01\cdot10^{-2}$ | $3.07\cdot10^{-4}$ | $5.37\cdot10^{-1}$ | $3.41\cdot10^{-3}$ |
| 29.500 | 0.000 | $2.67\cdot10^{-2}$ | $3.43\cdot10^{-4}$ | $4.45\cdot10^{-1}$ | $1.56\cdot10^{-3}$ | $1.11\cdot10^{-2}$ | $3.36\cdot10^{-4}$ | $5.33\cdot10^{-1}$ | $3.40\cdot10^{-3}$ |
| 29.500 | 2.375 | $2.50\cdot10^{-2}$ | $3.65\cdot10^{-4}$ | $4.49\cdot10^{-1}$ | $1.81\cdot10^{-3}$ | $1.08\cdot10^{-2}$ | $3.78\cdot10^{-4}$ | $5.32\cdot10^{-1}$ | $3.94\cdot10^{-3}$ |
| 29.500 | 7.125 | $2.14\cdot10^{-2}$ | $3.98\cdot10^{-4}$ | $4.60\cdot10^{-1}$ | $2.37\cdot10^{-3}$ | $9.12\cdot10^{-3}$ | $3.52\cdot10^{-4}$ | $5.46\cdot10^{-1}$ | $4.44\cdot10^{-3}$ |
| 29.500 | 11.875 | $1.88\cdot10^{-2}$ | $4.10\cdot10^{-4}$ | $4.69\cdot10^{-1}$ | $2.78\cdot10^{-3}$ | $7.71\cdot10^{-3}$ | $3.10\cdot10^{-4}$ | $5.61\cdot10^{-1}$ | $4.74\cdot10^{-3}$ |
| 29.500 | 16.625 | $1.62\cdot10^{-2}$ | $4.00\cdot10^{-4}$ | $4.82\cdot10^{-1}$ | $3.21\cdot10^{-3}$ | $7.10\cdot10^{-3}$ | $3.19\cdot10^{-4}$ | $5.66\cdot10^{-1}$ | $5.29\cdot10^{-3}$ |
| 29.500 | 19.000 | $1.53\cdot10^{-2}$ | $3.98\cdot10^{-4}$ | $4.86\cdot10^{-1}$ | $3.38\cdot10^{-3}$ | $6.86\cdot10^{-3}$ | $3.25\cdot10^{-4}$ | $5.68\cdot10^{-1}$ | $5.59\cdot10^{-3}$ |
| 29.500 | 21.250 | $1.46\cdot10^{-2}$ | $3.99\cdot10^{-4}$ | $4.90\cdot10^{-1}$ | $3.57\cdot10^{-3}$ | $6.65\cdot10^{-3}$ | $3.31\cdot10^{-4}$ | $5.69\cdot10^{-1}$ | $5.87\cdot10^{-3}$ |
| 29.500 | 25.750 | $1.33\cdot10^{-2}$ | $4.06\cdot10^{-4}$ | $4.98\cdot10^{-1}$ | $3.98\cdot10^{-3}$ | $5.69\cdot10^{-3}$ | $2.77\cdot10^{-4}$ | $5.86\cdot10^{-1}$ | $5.88\cdot10^{-3}$ |
| 29.500 | 30.250 | $1.16\cdot10^{-2}$ | $3.64\cdot10^{-4}$ | $5.12\cdot10^{-1}$ | $4.21\cdot10^{-3}$ | $5.39\cdot10^{-3}$ | $2.84\cdot10^{-4}$ | $5.89\cdot10^{-1}$ | $6.35\cdot10^{-3}$ |
| 29.500 | 34.750 | $1.06\cdot10^{-2}$ | $3.61\cdot10^{-4}$ | $5.19\cdot10^{-1}$ | $4.53\cdot10^{-3}$ | $5.14\cdot10^{-3}$ | $2.90\cdot10^{-4}$ | $5.92\cdot10^{-1}$ | $6.81\cdot10^{-3}$ |
| 29.500 | 39.250 | $9.84\cdot10^{-3}$ | $3.61\cdot10^{-4}$ | $5.26\cdot10^{-1}$ | $4.89\cdot10^{-3}$ | $4.41\cdot10^{-3}$ | $2.38\cdot10^{-4}$ | $6.10\cdot10^{-1}$ | $6.67\cdot10^{-3}$ |
| 29.500 | 43.750 | $9.17\cdot10^{-3}$ | $3.61\cdot10^{-4}$ | $5.32\cdot10^{-1}$ | $5.25\cdot10^{-3}$ | $4.23\cdot10^{-3}$ | $2.41\cdot10^{-4}$ | $6.13\cdot10^{-1}$ | $7.05\cdot10^{-3}$ |
| 29.500 | 48.250 | $8.04\cdot10^{-3}$ | $3.05\cdot10^{-4}$ | $5.47\cdot10^{-1}$ | $5.19\cdot10^{-3}$ | $4.08\cdot10^{-3}$ | $2.45\cdot10^{-4}$ | $6.15\cdot10^{-1}$ | $7.42\cdot10^{-3}$ |
| 29.500 | 52.750 | $7.55\cdot10^{-3}$ | $3.02\cdot10^{-4}$ | $5.53\cdot10^{-1}$ | $5.47\cdot10^{-3}$ | $3.96\cdot10^{-3}$ | $2.49\cdot10^{-4}$ | $6.17\cdot10^{-1}$ | $7.77\cdot10^{-3}$ |
| 29.500 | 57.250 | $7.13\cdot10^{-3}$ | $3.00\cdot10^{-4}$ | $5.58\cdot10^{-1}$ | $5.74\cdot10^{-3}$ | $3.40\cdot10^{-3}$ | $2.00\cdot10^{-4}$ | $6.37\cdot10^{-1}$ | $7.46\cdot10^{-3}$ |
| 29.500 | 61.750 | $6.78\cdot10^{-3}$ | $2.99\cdot10^{-4}$ | $5.62\cdot10^{-1}$ | $6.03\cdot10^{-3}$ | $3.31\cdot10^{-3}$ | $2.02\cdot10^{-4}$ | $6.39\cdot10^{-1}$ | $7.74\cdot10^{-3}$ |
| 29.500 | 66.250 | $6.47\cdot10^{-3}$ | $3.00\cdot10^{-4}$ | $5.66\cdot10^{-1}$ | $6.32\cdot10^{-3}$ | $3.23\cdot10^{-3}$ | $2.04\cdot10^{-4}$ | $6.40\cdot10^{-1}$ | $8.00\cdot10^{-3}$ |
| 29.500 | 70.750 | $5.70\cdot10^{-3}$ | $2.45\cdot10^{-4}$ | $5.83\cdot10^{-1}$ | $6.04\cdot10^{-3}$ | $3.16\cdot10^{-3}$ | $2.06\cdot10^{-4}$ | $6.42\cdot10^{-1}$ | $8.26\cdot10^{-3}$ |
| 29.500 | 75.250 | $5.46\cdot10^{-3}$ | $2.44\cdot10^{-4}$ | $5.87\cdot10^{-1}$ | $6.26\cdot10^{-3}$ | $3.11\cdot10^{-3}$ | $2.09\cdot10^{-4}$ | $6.43\cdot10^{-1}$ | $8.52\cdot10^{-3}$ |
| 29.500 | 79.750 | $5.25\cdot10^{-3}$ | $2.43\cdot10^{-4}$ | $5.90\cdot10^{-1}$ | $6.49\cdot10^{-3}$ | $3.06\cdot10^{-3}$ | $2.12\cdot10^{-4}$ | $6.43\cdot10^{-1}$ | $8.76\cdot10^{-3}$ |
| 29.500 | 84.250 | $5.07\cdot10^{-3}$ | $2.43\cdot10^{-4}$ | $5.94\cdot10^{-1}$ | $6.72\cdot10^{-3}$ | $3.03\cdot10^{-3}$ | $2.15\cdot10^{-4}$ | $6.44\cdot10^{-1}$ | $9.00\cdot10^{-3}$ |
| 29.500 | 88.750 | $4.91\cdot10^{-3}$ | $2.44\cdot10^{-4}$ | $5.96\cdot10^{-1}$ | $6.94\cdot10^{-3}$ | $2.60\cdot10^{-3}$ | $1.68\cdot10^{-4}$ | $6.65\cdot10^{-1}$ | $8.43\cdot10^{-3}$ |
| 29.500 | 91.000 | $4.84\cdot10^{-3}$ | $2.44\cdot10^{-4}$ | $5.98\cdot10^{-1}$ | $7.05\cdot10^{-3}$ | $2.58\cdot10^{-3}$ | $1.69\cdot10^{-4}$ | $6.65\cdot10^{-1}$ | $8.53\cdot10^{-3}$ |

| | | **SSD** | | | | **SSE** | | | |
|---|---|---|---|---|---|---|---|---|---|
| **New load** | **Preload** | $\boldsymbol{a}$ | $\sigma_{\boldsymbol{a}}$ | $\boldsymbol{b}$ | $\sigma_{\boldsymbol{b}}$ | $\boldsymbol{a}$ | $\sigma_{\boldsymbol{a}}$ | $\boldsymbol{b}$ | $\sigma_{\boldsymbol{b}}$ |
| [kPa] | [kPa] | [-] | [-] | [-] | [-] | [-] | [-] | [-] | [-] |
| 2.375 | 0.000 | $1.77\cdot10^{-3}$ | $5.87\cdot10^{-5}$ | $5.67\cdot10^{-1}$ | $3.70\cdot10^{-3}$ | $1.09\cdot10^{-3}$ | $6.11\cdot10^{-5}$ | $6.20\cdot10^{-1}$ | $6.43\cdot10^{-3}$ |
| 7.125 | 0.000 | $4.59\cdot10^{-3}$ | $9.82\cdot10^{-5}$ | $5.28\cdot10^{-1}$ | $2.47\cdot10^{-3}$ | $2.87\cdot10^{-3}$ | $1.22\cdot10^{-4}$ | $5.76\cdot10^{-1}$ | $4.62\cdot10^{-3}$ |
| 9.500 | 0.000 | $5.83\cdot10^{-3}$ | $9.33\cdot10^{-5}$ | $5.22\cdot10^{-1}$ | $1.90\cdot10^{-3}$ | $3.81\cdot10^{-3}$ | $1.60\cdot10^{-4}$ | $5.63\cdot10^{-1}$ | $4.58\cdot10^{-3}$ |
| 12.000 | 0.000 | $7.45\cdot10^{-3}$ | $1.15\cdot10^{-4}$ | $5.12\cdot10^{-1}$ | $1.84\cdot10^{-3}$ | $4.38\cdot10^{-3}$ | $1.59\cdot10^{-4}$ | $5.65\cdot10^{-1}$ | $4.05\cdot10^{-3}$ |
| 17.000 | 0.000 | $1.08\cdot10^{-2}$ | $1.37\cdot10^{-4}$ | $4.97\cdot10^{-1}$ | $1.56\cdot10^{-3}$ | $6.23\cdot10^{-3}$ | $2.34\cdot10^{-4}$ | $5.50\cdot10^{-1}$ | $4.21\cdot10^{-3}$ |
| 22.000 | 0.000 | $1.46\cdot10^{-2}$ | $1.82\cdot10^{-4}$ | $4.83\cdot10^{-1}$ | $1.59\cdot10^{-3}$ | $7.51\cdot10^{-3}$ | $2.49\cdot10^{-4}$ | $5.49\cdot10^{-1}$ | $3.82\cdot10^{-3}$ |
| 27.000 | 0.000 | $1.87\cdot10^{-2}$ | $2.41\cdot10^{-4}$ | $4.70\cdot10^{-1}$ | $1.64\cdot10^{-3}$ | $9.29\cdot10^{-3}$ | $3.11\cdot10^{-4}$ | $5.40\cdot10^{-1}$ | $3.86\cdot10^{-3}$ |
| 29.500 | 0.000 | $2.09\cdot10^{-2}$ | $2.75\cdot10^{-4}$ | $4.64\cdot10^{-1}$ | $1.68\cdot10^{-3}$ | $9.58\cdot10^{-3}$ | $2.82\cdot10^{-4}$ | $5.45\cdot10^{-1}$ | $3.48\cdot10^{-3}$ |
| 29.500 | 2.375 | $1.95\cdot10^{-2}$ | $2.97\cdot10^{-4}$ | $4.68\cdot10^{-1}$ | $1.99\cdot10^{-3}$ | $9.30\cdot10^{-3}$ | $3.20\cdot10^{-4}$ | $5.45\cdot10^{-1}$ | $4.07\cdot10^{-3}$ |
| 29.500 | 7.125 | $1.68\cdot10^{-2}$ | $3.41\cdot10^{-4}$ | $4.80\cdot10^{-1}$ | $2.74\cdot10^{-3}$ | $7.77\cdot10^{-3}$ | $2.96\cdot10^{-4}$ | $5.61\cdot10^{-1}$ | $4.63\cdot10^{-3}$ |
| 29.500 | 11.875 | $1.49\cdot10^{-2}$ | $3.74\cdot10^{-4}$ | $4.87\cdot10^{-1}$ | $3.36\cdot10^{-3}$ | $7.11\cdot10^{-3}$ | $3.21\cdot10^{-4}$ | $5.66\cdot10^{-1}$ | $5.46\cdot10^{-3}$ |
| 29.500 | 16.625 | $1.30\cdot10^{-2}$ | $3.59\cdot10^{-4}$ | $5.00\cdot10^{-1}$ | $3.83\cdot10^{-3}$ | $6.01\cdot10^{-3}$ | $2.68\cdot10^{-4}$ | $5.83\cdot10^{-1}$ | $5.55\cdot10^{-3}$ |
| 29.500 | 19.000 | $1.23\cdot10^{-2}$ | $3.66\cdot10^{-4}$ | $5.04\cdot10^{-1}$ | $4.10\cdot10^{-3}$ | $5.80\cdot10^{-3}$ | $2.74\cdot10^{-4}$ | $5.85\cdot10^{-1}$ | $5.87\cdot10^{-3}$ |
| 29.500 | 21.250 | $1.18\cdot10^{-2}$ | $3.72\cdot10^{-4}$ | $5.07\cdot10^{-1}$ | $4.35\cdot10^{-3}$ | $5.63\cdot10^{-3}$ | $2.80\cdot10^{-4}$ | $5.87\cdot10^{-1}$ | $6.19\cdot10^{-3}$ |
| 29.500 | 25.750 | $1.03\cdot10^{-2}$ | $3.31\cdot10^{-4}$ | $5.22\cdot10^{-1}$ | $4.56\cdot10^{-3}$ | $4.81\cdot10^{-3}$ | $2.29\cdot10^{-4}$ | $6.05\cdot10^{-1}$ | $6.07\cdot10^{-3}$ |
| 29.500 | 30.250 | $9.50\cdot10^{-3}$ | $3.32\cdot10^{-4}$ | $5.28\cdot10^{-1}$ | $4.96\cdot10^{-3}$ | $4.55\cdot10^{-3}$ | $2.34\cdot10^{-4}$ | $6.08\cdot10^{-1}$ | $6.56\cdot10^{-3}$ |
| 29.500 | 34.750 | $8.83\cdot10^{-3}$ | $3.35\cdot10^{-4}$ | $5.34\cdot10^{-1}$ | $5.37\cdot10^{-3}$ | $4.35\cdot10^{-3}$ | $2.41\cdot10^{-4}$ | $6.11\cdot10^{-1}$ | $7.06\cdot10^{-3}$ |
| 29.500 | 39.250 | $7.74\cdot10^{-3}$ | $2.82\cdot10^{-4}$ | $5.50\cdot10^{-1}$ | $5.31\cdot10^{-3}$ | $3.73\cdot10^{-3}$ | $1.92\cdot10^{-4}$ | $6.30\cdot10^{-1}$ | $6.76\cdot10^{-3}$ |
| 29.500 | 43.750 | $7.25\cdot10^{-3}$ | $2.80\cdot10^{-4}$ | $5.55\cdot10^{-1}$ | $5.62\cdot10^{-3}$ | $3.58\cdot10^{-3}$ | $1.96\cdot10^{-4}$ | $6.33\cdot10^{-1}$ | $7.15\cdot10^{-3}$ |
| 29.500 | 48.250 | $6.84\cdot10^{-3}$ | $2.79\cdot10^{-4}$ | $5.60\cdot10^{-1}$ | $5.94\cdot10^{-3}$ | $3.46\cdot10^{-3}$ | $1.99\cdot10^{-4}$ | $6.36\cdot10^{-1}$ | $7.54\cdot10^{-3}$ |
| 29.500 | 52.750 | $6.49\cdot10^{-3}$ | $2.79\cdot10^{-4}$ | $5.64\cdot10^{-1}$ | $6.26\cdot10^{-3}$ | $3.36\cdot10^{-3}$ | $2.03\cdot10^{-4}$ | $6.37\cdot10^{-1}$ | $7.92\cdot10^{-3}$ |
| 29.500 | 57.250 | $6.18\cdot10^{-3}$ | $2.80\cdot10^{-4}$ | $5.67\cdot10^{-1}$ | $6.59\cdot10^{-3}$ | $3.27\cdot10^{-3}$ | $2.07\cdot10^{-4}$ | $6.39\cdot10^{-1}$ | $8.29\cdot10^{-3}$ |
| 29.500 | 61.750 | $5.46\cdot10^{-3}$ | $2.28\cdot10^{-4}$ | $5.85\cdot10^{-1}$ | $6.26\cdot10^{-3}$ | $2.81\cdot10^{-3}$ | $1.62\cdot10^{-4}$ | $6.60\cdot10^{-1}$ | $7.74\cdot10^{-3}$ |
| 29.500 | 66.250 | $5.23\cdot10^{-3}$ | $2.27\cdot10^{-4}$ | $5.88\cdot10^{-1}$ | $6.50\cdot10^{-3}$ | $2.74\cdot10^{-3}$ | $1.64\cdot10^{-4}$ | $6.61\cdot10^{-1}$ | $8.02\cdot10^{-3}$ |
| 29.500 | 70.750 | $5.02\cdot10^{-3}$ | $2.26\cdot10^{-4}$ | $5.91\cdot10^{-1}$ | $6.75\cdot10^{-3}$ | $2.69\cdot10^{-3}$ | $1.66\cdot10^{-4}$ | $6.62\cdot10^{-1}$ | $8.29\cdot10^{-3}$ |
| 29.500 | 75.250 | $4.84\cdot10^{-3}$ | $2.26\cdot10^{-4}$ | $5.94\cdot10^{-1}$ | $6.99\cdot10^{-3}$ | $2.65\cdot10^{-3}$ | $1.68\cdot10^{-4}$ | $6.63\cdot10^{-1}$ | $8.54\cdot10^{-3}$ |
| 29.500 | 79.750 | $4.69\cdot10^{-3}$ | $2.26\cdot10^{-4}$ | $5.97\cdot10^{-1}$ | $7.22\cdot10^{-3}$ | $2.61\cdot10^{-3}$ | $1.71\cdot10^{-4}$ | $6.64\cdot10^{-1}$ | $8.80\cdot10^{-3}$ |
| 29.500 | 84.250 | $4.55\cdot10^{-3}$ | $2.26\cdot10^{-4}$ | $5.99\cdot10^{-1}$ | $7.44\cdot10^{-3}$ | $2.58\cdot10^{-3}$ | $1.74\cdot10^{-4}$ | $6.64\cdot10^{-1}$ | $9.04\cdot10^{-3}$ |
| 29.500 | 88.750 | $4.05\cdot10^{-3}$ | $1.83\cdot10^{-4}$ | $6.16\cdot10^{-1}$ | $6.97\cdot10^{-3}$ | $2.56\cdot10^{-3}$ | $1.76\cdot10^{-4}$ | $6.64\cdot10^{-1}$ | $9.27\cdot10^{-3}$ |
| 29.500 | 91.000 | $3.99\cdot10^{-3}$ | $1.82\cdot10^{-4}$ | $6.18\cdot10^{-1}$ | $7.05\cdot10^{-3}$ | $2.55\cdot10^{-3}$ | $1.78\cdot10^{-4}$ | $6.64\cdot10^{-1}$ | $9.38\cdot10^{-3}$ |

## A.2. Calculated Features 1 and 2

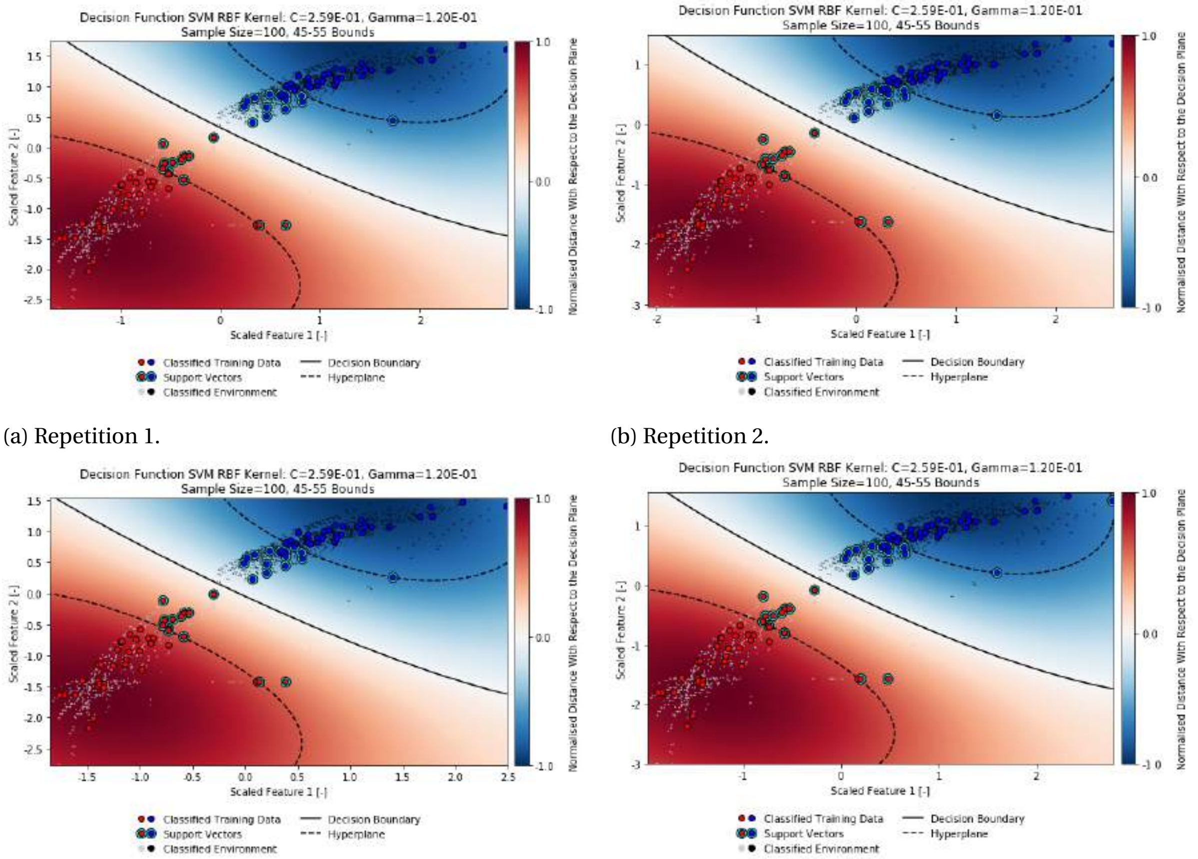
Table A.2: Calculated features by the constructed feature extraction method from the WBI-SOS subsoil scenarios (without layer interface distributions), each subjected to 30 load combinations.

| | | SSA | | SSC | | SSD | | SSE | |
|---|---|---|---|---|---|---|---|---|---|
| **New load** | **Preload** | **Feature 1** | **Feature 2** | **Feature 1** | **Feature 2** | **Feature 1** | **Feature 2** | **Feature 1** | **Feature 2** |
| [kPa] | [kPa] | [-] | [-] | [-] | [-] | [-] | [-] | [-] | [-] |
| 2.375 | 0.000 | $6.96\cdot10^{-4}$ | $7.56\cdot10^{-2}$ | $4.06\cdot10^{-4}$ | $3.85\cdot10^{-2}$ | $6.88\cdot10^{-4}$ | $7.98\cdot10^{-2}$ | $4.21\cdot10^{-4}$ | $3.85\cdot10^{-2}$ |
| 7.125 | 0.000 | $6.21\cdot10^{-4}$ | $8.41\cdot10^{-2}$ | $3.60\cdot10^{-4}$ | $5.27\cdot10^{-2}$ | $5.39\cdot10^{-4}$ | $8.07\cdot10^{-2}$ | $3.37\cdot10^{-4}$ | $4.74\cdot10^{-2}$ |
| 9.500 | 0.000 | $6.06\cdot10^{-4}$ | $8.26\cdot10^{-2}$ | $3.28\cdot10^{-4}$ | $4.68\cdot10^{-2}$ | $5.00\cdot10^{-4}$ | $7.64\cdot10^{-2}$ | $3.27\cdot10^{-4}$ | $4.99\cdot10^{-2}$ |
| 12.000 | 0.000 | $6.22\cdot10^{-4}$ | $8.36\cdot10^{-2}$ | $3.25\cdot10^{-4}$ | $4.86\cdot10^{-2}$ | $4.95\cdot10^{-4}$ | $7.57\cdot10^{-2}$ | $2.91\cdot10^{-4}$ | $4.00\cdot10^{-2}$ |
| 17.000 | 0.000 | $6.34\cdot10^{-4}$ | $8.26\cdot10^{-2}$ | $2.99\cdot10^{-4}$ | $4.33\cdot10^{-2}$ | $4.90\cdot10^{-4}$ | $7.36\cdot10^{-2}$ | $2.83\cdot10^{-4}$ | $4.25\cdot10^{-2}$ |
| 22.000 | 0.000 | $6.53\cdot10^{-4}$ | $8.24\cdot10^{-2}$ | $2.96\cdot10^{-4}$ | $4.43\cdot10^{-2}$ | $4.99\cdot10^{-4}$ | $7.32\cdot10^{-2}$ | $2.57\cdot10^{-4}$ | $3.53\cdot10^{-2}$ |
| 27.000 | 0.000 | $6.59\cdot10^{-4}$ | $8.13\cdot10^{-2}$ | $2.77\cdot10^{-4}$ | $3.99\cdot10^{-2}$ | $5.13\cdot10^{-4}$ | $7.36\cdot10^{-2}$ | $2.55\cdot10^{-4}$ | $3.72\cdot10^{-2}$ |
| 29.500 | 0.000 | $6.64\cdot10^{-4}$ | $8.10\cdot10^{-2}$ | $2.76\cdot10^{-4}$ | $4.04\cdot10^{-2}$ | $5.19\cdot10^{-4}$ | $7.37\cdot10^{-2}$ | $2.39\cdot10^{-4}$ | $3.07\cdot10^{-2}$ |
| 29.500 | 2.375 | $6.02\cdot10^{-4}$ | $8.04\cdot10^{-2}$ | $2.61\cdot10^{-4}$ | $3.10\cdot10^{-2}$ | $4.71\cdot10^{-4}$ | $6.83\cdot10^{-2}$ | $2.25\cdot10^{-4}$ | $1.97\cdot10^{-2}$ |
| 29.500 | 7.125 | $6.85\cdot10^{-4}$ | $8.29\cdot10^{-2}$ | $2.89\cdot10^{-4}$ | $4.05\cdot10^{-2}$ | $5.36\cdot10^{-4}$ | $7.25\cdot10^{-2}$ | $2.45\cdot10^{-4}$ | $3.19\cdot10^{-2}$ |
| 29.500 | 11.875 | $7.12\cdot10^{-4}$ | $8.50\cdot10^{-2}$ | $2.86\cdot10^{-4}$ | $4.27\cdot10^{-2}$ | $5.63\cdot10^{-4}$ | $7.61\cdot10^{-2}$ | $2.63\cdot10^{-4}$ | $4.02\cdot10^{-2}$ |
| 29.500 | 16.625 | $6.99\cdot10^{-4}$ | $8.42\cdot10^{-2}$ | $2.97\cdot10^{-4}$ | $4.75\cdot10^{-2}$ | $5.53\cdot10^{-4}$ | $7.59\cdot10^{-2}$ | $2.50\cdot10^{-4}$ | $3.83\cdot10^{-2}$ |
| 29.500 | 19.000 | $6.97\cdot10^{-4}$ | $8.46\cdot10^{-2}$ | $3.02\cdot10^{-4}$ | $4.96\cdot10^{-2}$ | $5.55\cdot10^{-4}$ | $7.69\cdot10^{-2}$ | $2.54\cdot10^{-4}$ | $4.07\cdot10^{-2}$ |
| 29.500 | 21.250 | $6.96\cdot10^{-4}$ | $8.50\cdot10^{-2}$ | $3.07\cdot10^{-4}$ | $5.14\cdot10^{-2}$ | $5.57\cdot10^{-4}$ | $7.77\cdot10^{-2}$ | $2.58\cdot10^{-4}$ | $4.26\cdot10^{-2}$ |
| 29.500 | 25.750 | $6.92\cdot10^{-4}$ | $8.57\cdot10^{-2}$ | $2.84\cdot10^{-4}$ | $4.82\cdot10^{-2}$ | $5.31\cdot10^{-4}$ | $7.57\cdot10^{-2}$ | $2.38\cdot10^{-4}$ | $3.60\cdot10^{-2}$ |
| 29.500 | 30.250 | $6.54\cdot10^{-4}$ | $8.32\cdot10^{-2}$ | $2.91\cdot10^{-4}$ | $5.08\cdot10^{-2}$ | $5.30\cdot10^{-4}$ | $7.67\cdot10^{-2}$ | $2.43\cdot10^{-4}$ | $3.93\cdot10^{-2}$ |
| 29.500 | 34.750 | $6.47\cdot10^{-4}$ | $8.34\cdot10^{-2}$ | $2.98\cdot10^{-4}$ | $5.32\cdot10^{-2}$ | $5.31\cdot10^{-4}$ | $7.77\cdot10^{-2}$ | $2.49\cdot10^{-4}$ | $4.23\cdot10^{-2}$ |
| 29.500 | 39.250 | $6.43\cdot10^{-4}$ | $8.37\cdot10^{-2}$ | $2.71\cdot10^{-4}$ | $4.72\cdot10^{-2}$ | $4.97\cdot10^{-4}$ | $7.43\cdot10^{-2}$ | $2.26\cdot10^{-4}$ | $3.15\cdot10^{-2}$ |
| 29.500 | 43.750 | $6.42\cdot10^{-4}$ | $8.42\cdot10^{-2}$ | $2.77\cdot10^{-4}$ | $4.91\cdot10^{-2}$ | $4.97\cdot10^{-4}$ | $7.50\cdot10^{-2}$ | $2.31\cdot10^{-4}$ | $3.23\cdot10^{-2}$ |
| 29.500 | 48.250 | $5.98\cdot10^{-4}$ | $8.05\cdot10^{-2}$ | $2.84\cdot10^{-4}$ | $5.11\cdot10^{-2}$ | $5.00\cdot10^{-4}$ | $7.57\cdot10^{-2}$ | $2.37\cdot10^{-4}$ | $3.31\cdot10^{-2}$ |
| 29.500 | 52.750 | $5.98\cdot10^{-4}$ | $8.07\cdot10^{-2}$ | $2.92\cdot10^{-4}$ | $5.31\cdot10^{-2}$ | $5.05\cdot10^{-4}$ | $7.65\cdot10^{-2}$ | $2.44\cdot10^{-4}$ | $3.38\cdot10^{-2}$ |
| 29.500 | 57.250 | $6.02\cdot10^{-4}$ | $8.11\cdot10^{-2}$ | $2.63\cdot10^{-4}$ | $4.26\cdot10^{-2}$ | $5.12\cdot10^{-4}$ | $7.74\cdot10^{-2}$ | $2.52\cdot10^{-4}$ | $4.00\cdot10^{-2}$ |
| 29.500 | 61.750 | $6.08\cdot10^{-4}$ | $8.15\cdot10^{-2}$ | $2.70\cdot10^{-4}$ | $4.48\cdot10^{-2}$ | $4.76\cdot10^{-4}$ | $7.28\cdot10^{-2}$ | $2.25\cdot10^{-4}$ | $3.52\cdot10^{-2}$ |
| 29.500 | 66.250 | $6.18\cdot10^{-4}$ | $8.20\cdot10^{-2}$ | $2.80\cdot10^{-4}$ | $4.69\cdot10^{-2}$ | $4.84\cdot10^{-4}$ | $7.34\cdot10^{-2}$ | $2.32\cdot10^{-4}$ | $3.58\cdot10^{-2}$ |
| 29.500 | 70.750 | $5.73\cdot10^{-4}$ | $7.74\cdot10^{-2}$ | $2.90\cdot10^{-4}$ | $4.91\cdot10^{-2}$ | $4.94\cdot10^{-4}$ | $7.42\cdot10^{-2}$ | $2.41\cdot10^{-4}$ | $3.64\cdot10^{-2}$ |
| 29.500 | 75.250 | $5.84\cdot10^{-4}$ | $7.78\cdot10^{-2}$ | $3.02\cdot10^{-4}$ | $5.12\cdot10^{-2}$ | $5.07\cdot10^{-4}$ | $7.49\cdot10^{-2}$ | $2.50\cdot10^{-4}$ | $3.70\cdot10^{-2}$ |
| 29.500 | 79.750 | $5.99\cdot10^{-4}$ | $7.82\cdot10^{-2}$ | $3.16\cdot10^{-4}$ | $5.32\cdot10^{-2}$ | $5.22\cdot10^{-4}$ | $7.58\cdot10^{-2}$ | $2.61\cdot10^{-4}$ | $3.75\cdot10^{-2}$ |
| 29.500 | 84.250 | $6.17\cdot10^{-4}$ | $7.88\cdot10^{-2}$ | $3.31\cdot10^{-4}$ | $5.53\cdot10^{-2}$ | $5.41\cdot10^{-4}$ | $7.67\cdot10^{-2}$ | $2.74\cdot10^{-4}$ | $3.80\cdot10^{-2}$ |
| 29.500 | 88.750 | $6.40\cdot10^{-4}$ | $7.93\cdot10^{-2}$ | $2.93\cdot10^{-4}$ | $3.85\cdot10^{-2}$ | $5.03\cdot10^{-4}$ | $7.10\cdot10^{-2}$ | $2.88\cdot10^{-4}$ | $3.85\cdot10^{-2}$ |
| 29.500 | 91.000 | $6.53\cdot10^{-4}$ | $7.97\cdot10^{-2}$ | $3.00\cdot10^{-4}$ | $3.88\cdot10^{-2}$ | $5.13\cdot10^{-4}$ | $7.14\cdot10^{-2}$ | $2.95\cdot10^{-4}$ | $3.88\cdot10^{-2}$ |

# B

# TRAINING RESULTS VISUALISATIONS

## B.1. Training Results on 100 Training Data Points Sampled Between the 45 - 55 Sample Bounds



(a) Repetition 1.

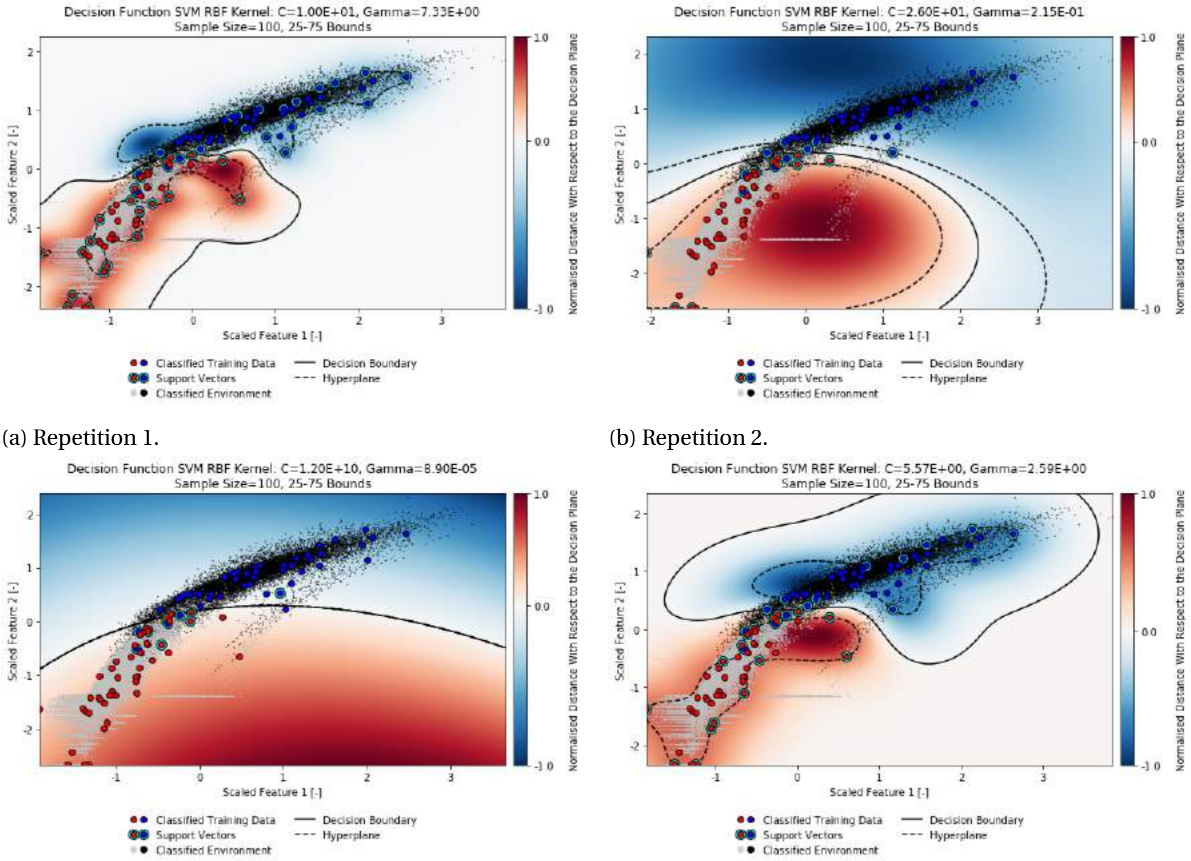
(b) Repetition 2.

(c) Repetition 3.

(d) Repetition 4.

Figure B.1: Visualisations of the results of four training processes, trained on a single training data sample of 100 training data points, sampled between the 45-55 sample bounds.

## B.2. Training Results on 100 Training Data Points Sampled Between the 25 - 75 Sample Bounds



(a) Repetition 1.

(b) Repetition 2.

(c) Repetition 3.

(d) Repetition 4.

Figure B.2: Visualisations of the results of four training processes trained on a single training data sample of 100 training data points, sampled between the 25-75 sample bounds.